Cognos® 8 Business Intelligence

REPORT STUDIO

PROFESSIONAL AUTHORING USER GUIDE

COGNOS®

THE NEXT LEVEL OF PERFORMANCE™

Product Information

This document applies to Cognos® 8 Version 8.3 and may also apply to subsequent releases. To check for newer versions of this document, visit the Cognos Global Customer Services Web site (http://support.cognos.com).

Copyright

Copyright © 2007 Cognos Incorporated.

Portions of Cognos® software products are protected by one or more of the following U.S. Patents: 6,609,123 B1; 6,611,838 B1; 6,662,188 B1; 6,728,697 B2; 6,741,982 B2; 6,763,520 B1; 6,768,995 B2; 6,782,378 B2; 6,847,973 B2; 6,907,428 B2; 6,853,375 B2; 6,986,135 B2; 6,995,768 B2; 7,062,479 B2; 7,072,822 B2; 7,111,007 B2; 7,130,822 B1; 7,155,398 B2; 7,171,425 B2; 7,185,016 B1;7,213,199 B2.

Cognos and the Cognos logo are trademarks of Cognos Incorporated in the United States and/or other countries. All other names are trademarks or registered trademarks of their respective companies.

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. Cognos does not accept responsibility for any kind of loss resulting from the use of information contained in this document.

This document shows the publication date. The information contained in this document is subject to change without notice. Any improvements or changes to either the product or the document will be documented in subsequent editions.

U.S. Government Restricted Rights. The software and accompanying materials are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to the restrictions in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (C)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48CFR52.227-19, as applicable. The Contractor is Cognos Corporation, 15 Wayside Road, Burlington, MA 01803.

This software/documentation contains proprietary information of Cognos Incorporated. All rights are reserved. Reverse engineering of this software is prohibited. No part of this software/documentation may be copied, photocopied, reproduced, stored in a retrieval system, transmitted in any form or by any means, or translated into another language without the prior written consent of Cognos Incorporated.

Table of Contents

Introduction 37

Chapter 1: What's New? 39

New Features in Version 8.3 39

Professional and Express Authoring Modes 39

Enhanced User Interface When Working With Dimensional Data Sources 39

Enhanced Conditional Styles 40

Extended Crosstab Indentation 40

Customizable Data Tree 40

Table of Contents in PDF Reports 40

No Data Handling 40

Simplified Report Style 41

Enhanced Report Expressions 41

Singleton Results in Layout 42

PDF Horizontal Pagination 42

Page-by-Page PDF Page Layout 43

Zooming on Chart Region 43

Other New Features 43

Changed Features in Version 8.3 44

Product Behavior After Upgrade 44

Special Cell Values Shown Differently 44

Slicer Filters Now Affect All Edge Values 44

Default Calculation Position 45

Changes to Default Data Formats for OLAP Data Sources 45

Summaries on Non-Additive Measures Give Incorrect Results for SAP BW 46

FOR Aggregate is No Longer Context-dependent for OLAP 46

Rounding with Equality Filters for MSAS Cubes 47

Essbase Calculations on Missing Facts 47

Changes to the Data Tree for SAP BW 47

Deprecated Features in Version 8.3 48

Excel 2000 Format 48

Chapter 2: Report Studio 49

Planning Reports 49

Building Cognos 8 Applications 49

Working in Report Studio 50

The User Interface 51

Basic Report Structure 54

Working with Objects 56

Set Options 57

Web Browser Settings 61

Authoring Modes 61

Creating a Report 62

Specify the Package 62

Choose a Report Template 63
Add Data to a Report 64
Save a Report 66
Run a Report 66
Set PDF Page Options 73
Creating Your Own Report Templates 74
Convert a Report to a Template 74
Create a New Template 75
Techniques for Creating Accessible Reports 75
Techniques for Creating Reports for Cognos 8 Go! Office 76
Managing Your Reports 77
The Cognos 8 SDK 78

Chapter 3: Types of Reports 79
List Reports 79
Group Data 80
Set the Group Span for a Column 81
Format Lists 84
Crosstab Reports 85
Crosstab Nodes and Crosstab Node Members 86
Create a Single-Edge Crosstab Report 87
Create a Nested Crosstab Report 88
Format Crosstabs 89
Indent Data 91
Change a List into a Crosstab 91
Example - Add Aggregate Data to a Crosstab Report 92
Example - Create a Discontinuous Crosstab 93
Charts 94
Example - Create a Column Chart to Plot Order Trends 95
Customize a Chart 96
Customize the Colors of a Chart 99
Change a Chart Background 101
Add a Baseline to a Chart 102
Customize the Lines in a Chart 105
Add a Marker to a Chart 106
Add a Note to a Chart 107
Specify the Chart Label Text and Value Text Shown 111
Specify Custom Label Text 111
Specify the Properties of a Gauge Chart 112
Create a Drill-up and Drill-down Chart 112
Example - Show Values in the Chart Legend 113
Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels 115
Example - Customize the Axis Titles 115
Example - Create a Conditional Palette 117
Maps 119
Example - Create a Map Report 120
Example - Define Data Values for the Region Layer 121
Ignore Data with no Features 122
Match Data Values to Names in the Map File 122

Define Data Values for the Point Layer *123*
Add Another Color to the Region or Point Layer *124*
Specify the Size of Points *124*
Add Titles, Subtitles, Footers, and Axis Title *125*
Add Legend Titles *125*
Add a Note to a Map *126*
Customize a Map *126*
Drill Through to Another Report From a Map *128*
Edit a Map *129*
Additional Maps *129*
Location Intelligence *129*
Repeaters *130*
Convert a List into a Repeater *130*
Example - Create Mailing Labels *131*

Chapter 4: Formatting a Report *133*
Recommendation - Laying Out a Report *134*
Report Layout Guidelines *134*
The Page Structure View *134*
Add a Header or Footer *135*
Add Borders *137*
Add Text *137*
Specify the Font *138*
Add Color *138*
Copy and Reuse Formatting *139*
Insert an Image *139*
Insert a Background Image *139*
Create a Table of Contents *140*
Add a Bookmark *141*
Insert Other Objects *143*
Elements Supported in Rich Text Items *145*
Example - Add a Multimedia File to a Report *146*
Align an Object *147*
Use Tables to Control Where Objects Appear *147*
Apply a Table Style *147*
Apply Padding *148*
Set Margins *148*
Reuse a Layout Object *149*
Change a Reused Object *150*
Update Reused Objects *151*
Add a Page *151*
Create Page Sets *152*
Join Nested Page Sets *153*
Example - Preparing a Product List Report *154*
Reorder Rows and Columns *155*
Rename a Row or Column *156*
Swap Columns and Rows *157*
Insert Page Numbers *157*
Setting Object Properties *158*

Control Page Breaks and Page Numbering 158
Specify Text Properties 160
Specify Text Flow 160
Specify the Height and Width 161
Control How Objects Flow Around Other Objects 162
Specify Borders and Cell Size in Tables 163
Specify Report Properties 164
Create and Modify Classes 164
Modify Classes to Format Query Studio Reports 165
Modifying the Default Layout Style Sheet 166

Chapter 5: Working with Data 167
Working with Dimensional Data 167
Insert Members from the Tree 169
Create a Set of Members 169
Insert a Hierarchy 170
Insert a Member Property 170
Nest Members 170
Search for Members 171
Filter Data 172
Filter Values Using Context 174
Filtering Data Using an SAP BW Data Source 175
Ignoring the Time Component in Date Columns 176
Sort Data 177
Perform Advanced Sorting 179
Create Sections 180
Remove Sections 181
Create Page Layers 181
Summarizing Data 182
Use the Aggregation Properties Set in the Model to Specify Detail or Summary Aggregation 183
Set the Auto Group & Summarize Property 183
Use the Aggregate Button to Add Simple Summaries 184
Specify Aggregation Properties for a Data Item 185
Use Aggregate Functions in Data Item Expressions 185
Specify the Aggregation Mode 186
Aggregating Values in Crosstabs and Charts 186
Aggregating Member Sets 188
Create a Calculation 189
Insert a Calculated Member or Measure 190
Create an Intersection (Tuple) 191
Multiple Calculations in Crosstabs and Charts 191
Format Data 193
Specify the Format for an Object 195
Specifying the Number of Decimals 195
Locale-sensitive Properties 195
Insert a Single Data Item 196
Specify the List of Data Items for an Object 197
Show Data for a Specific Time Period 197

Specify the Default Measure 198
Specify What Appears for a Data Container that Contains No Data 199
Adding Prompts 199
Define Prompts Using Context Filters 200
Use the Build Prompt Page Tool 200
Build Your Own Prompt and Prompt Page 201
Create a Parameter to Produce a Prompt 205
Create a Prompt Directly in a Report Page 206
Modifying Prompts 207
Control the Data That Appears in a Tree Prompt 212
Working with Queries 213
Relating Queries to Layouts 214
Working with Dimensional Data Sources 214
Connecting Queries Between Different Data Sources 214
Multiple-fact Queries 215
Add Queries to a Report 217
Create a Union Query 219
Create a Join Relationship 222
Add Dimension Information to a Query 223
Reference Package Items in Child Queries 226
Working with SQL or MDX 227
Add a Query Macro 231

Chapter 6: Working with Existing Reports 235
Open and Save a Report Locally 235
Open a Report from the Clipboard 236
Open Files from Other Studios 236
An Analysis Studio Query Specification 237
Managing Changes in the Package 242
Update Name References 242
Change the Package 243
Add Multiple Items to a Single Column 243
Example - Create a Report with Multiple Items in One Column 244
Bursting Reports 246
Defining Burst Recipients 246
Specify Burst Groups 249
Set Burst Options 249
Enable Bursting 251
Creating Burst Reports Using a Dimensional Data Source 251
Example - Burst a Report 252
Customizing Reports Using Conditions 254
Highlight Data Using Conditional Styles 254
Highlight Data Using Style Variables 257
Adding Conditional Rendering to a Report 258
Drill-through Access 267
Understanding Drill-through Concepts 268
Set Up Drill-through Access in a Report 273
Create a Drill-up/Drill-down Report 279
Member Sets 281

Create a Master Detail Relationship 282
Link Members from Two Dimensional Data Sources 283

Chapter 7: Try It Yourself Exercises 285
Try It Yourself - Create a List Report 285
Try It Yourself - Create a Crosstab 287
Try It Yourself - Create Charts 288
Try It Yourself - Create Map Reports 298
Show the Distribution of Revenue by Country 298
Show Revenue and Margins for Cities in the United States 299
Try It Yourself - Create a Multiple-Page Report 301
Try It Yourself - Create a Dashboard Report 303
Try It Yourself - Create a Dynamic Report That Shows Period-to-date Data 305
Try It Yourself - Create a Report with Drill-through Access to Itself 307

Chapter 8: Upgrading Reports 313
Limitations When Upgrading Reports from Cognos ReportNet 313
Changes in the Behavior of Count Between Cognos ReportNet(R) 1.1 and Cognos 8 314
Upgrading Reports from Cognos 8 Version 8.1 or 8.2 315
Product Behavior After Upgrade 315
Members Containing No Data Are Not Suppressed for SAP BW 315
Thousands Separators Missing 315

Appendix A: Troubleshooting 317
Problems Creating Reports 317
Division by Zero Operation Appears Differently in Lists and Crosstabs 317
Application Error Appears When Upgrading a Report 317
Background Color in Template Does not Appear 317
Subtotals in Grouped Lists 318
Chart Labels Overwrite One Another 318
Chart Shows Only Every Second Label 318
Chart Gradient Backgrounds Appear Gray 318
Problems Filtering Data 318
HRESULT= DB_E_CANTCONVERTVALUE Error When Filtering on a _make_timestamp Column 319
Problems Running Reports 319
Summaries in Report Do not Correspond to the Visible Members 319
Cannot Find the Database in the Content Store (Error QE-DEF-0288) 320
Parse Errors When Opening or Running an Upgraded Report 321
Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters 321
The ORA-00907 Error Appears When Running a Report 321
A Report or Analysis Does Not Run Because of Missing Items 321
Cannot View Burst Report 321
A Report Upgraded from ReportNet Does Not Retain its Original Look 322
Drill-through Links Not Active in the Safari Browser 323
A Running Total in Grouped Reports Gives Unexpected Results 323
Problems When Drilling Through 323
Cannot Drill Between PowerCubes Because MUNs Do Not Match 323
Unexpected Results When Drilling from a Cube to a Relational Data Source 324
Cannot Drill From a Relational Source to a Cube 324

Appendix B: Samples 327
Samples in the GO Data Warehouse (analysis) Package 327
Customer Returns and Satisfaction 327
Employee Training by Year 327
Great Outdoors Company Balance Sheet as at Dec 31, 2006 328
Global Bonus Report 328
Succession Report 328
Manager Profile 328
Planned Headcount 329
Positions to Fill 329
Recruitment Report 329
Returned Items 329
Returns by Failed Orders in 2006 329
Returns by Damage, Failed Orders and Complaints in 2006 330
Revenue by GO Subsidiary 2005 330
Returns by Order Method - Prompted Chart 330
Samples in the GO Data Warehouse (query) Package 331
Bursted Sales Performance Report 331
Samples in the GO Sales (query) Package 331
Order Invoices - Donald Chow, Sales Person 331
Great Outdoors Samples 331
The Great Outdoors Group of Companies 332
Employees 334
Sales and Marketing 334
Great Outdoors Database, Models, and Packages 335
Appendix C: Chart Types 337
Choosing a Chart Type and Configuration 338
Pie Charts 339
Column Charts 339
Progressive Column Charts 340
Pareto Charts 340
Bar Charts 341
Line Charts 341
Area Charts 342
Combination Charts 342
Radar Charts 343
Scatter Charts 343
Bubble Charts 344
Point Charts 344
Quadrant Charts 345
Polar Charts 345
Metrics Range Charts 346
Gauge Charts 346
Chart Configurations 347
Standard Charts 347
Stacked Charts 348
100% Stacked Charts 348
3-D Charts 349

Appendix D: Summary Functions 351
Mapping Aggregation Functions From Data Sources to Framework Manager and Report Studio 355
Appendix E: Using the Expression Editor 357
Creating Expressions Using SAP BW Data Sources 358
Browse the Data of a Data Item 359
Searching for Values May Return Unexpected Results 359
Using Quotation Marks in Literal Strings 359
Recommendation - Use Member Unique Name (MUN) Aliases 360
Running a Report That Contains Members 360
Functions Not Available When Creating a Report or Layout Expression 360
Concatenating Strings 361
Dimensional Coercion Rules 361
Coercion Rules for Dimensional Objects 363
Coercion Rules for Numeric and String Values 364
Coercion Paths 364
Operators 365
(365
) 365
* 366
, 366
/ 366
|| 366
+ 366
- 366
< 367
<= 367
<> 367
= 367
> 367
-> 367
>= 368
and 368
auto 368
between 368
case 368
contains 368
currentMeasure 369
default 369
distinct 369
else 369
end 369
ends with 369
for 370
for all 370
for any 370
for report 370
if 370

in *370*
in_range *371*
is missing *371*
is null *371*
is not missing *371*
is not null *371*
like *371*
lookup *372*
not *372*
or *372*
prefilter *372*
rows *372*
starts with *372*
then *373*
when *373*
Summaries *373*
aggregate *373*
average *373*
count *374*
maximum *374*
median *374*
minimum *374*
moving-average *375*
moving-total *375*
percentage *376*
percentile *377*
quantile *377*
quartile *378*
rank *379*
running-average *380*
running-count *380*
running-difference *381*
running-maximum *381*
running-minimum *382*
running-total *383*
standard-deviation *383*
standard-deviation-pop *384*
total *384*
variance *384*
variance-pop *385*
Member Summaries *385*
aggregate *385*
average *385*
count *386*
maximum *386*
median *386*
minimum *386*
percentage *386*
percentile *387*

quantile *387*
quartile *387*
rank *387*
standard-deviation *388*
total *388*
variance *388*
Constants *388*
date *388*
date-time *388*
time with time zone *388*
timestamp with time zone *389*
interval *389*
interval year *389*
interval month *389*
interval year to month *389*
interval day *389*
interval hour *389*
interval minute *389*
interval second *389*
interval day to hour *389*
interval day to minute *389*
interval day to second *389*
interval hour to minute *390*
interval hour to second *390*
interval minute to second *390*
null *390*
number *390*
string *390*
time *390*
Constructs *390*
if then else *390*
in_range *390*
search case *391*
simple case *391*
Business Date/Time Functions *391*
_add_days *391*
_add_months *392*
_add_years *392*
_age *392*
_day_of_week *393*
_day_of_year *393*
_days_between *393*
_days_to_end_of_month *394*
_first_of_month *394*
_last_of_month *394*
_make_timestamp *395*
_months_between *395*
_week_of_year *395*
_years_between *395*

_ymdint_between 396
Block Functions 396
_firstFromSet 396
_remainderSet 396
Macro Functions 397
+ 397
array 397
csv 397
dq 398
grep 398
join 398
prompt 399
promptmany 399
sb 400
sq 400
sort 400
split 400
substitute 401
unique 401
urlencode 402
CSVIdentityName 402
CSVIdentityNameList 402
CAMPassport 403
CAMIDList 403
CAMIDListForType 403
Common Functions 404
nullif 404
_format 404
_round 404
abs 404
ancestor 405
ancestors 405
bottomCount 405
bottomPercent 405
bottomSum 405
caption 406
cast 406
ceil 407
ceiling 407
char_length 407
character_length 407
children 408
closingPeriod 408
coalesce 408
completeTuple 408
cousin 408
current_date 409
current_time 409
current_timestamp 409

currentMember *409*
defaultMember *410*
descendants *410*
emptySet *410*
except *411*
exp *411*
extract *411*
filter *411*
firstChild *412*
firstSibling *412*
floor *412*
generate *412*
head *412*
hierarchize *413*
hierarchy *413*
intersect *413*
item *413*
lag *413*
lastChild *413*
lastPeriods *414*
lastSibling *414*
lead *414*
level *414*
levels *414*
linkMember *414*
ln *415*
localtime *415*
localtimestamp *415*
lower *415*
member *416*
members *416*
mod *416*
nestedSet *416*
nextMember *417*
octet_length *417*
openingPeriod *417*
order *417*
ordinal *418*
parallelPeriod *418*
parent *418*
periodsToDate *418*
position *418*
power *419*
prevMember *419*
roleValue *419*
rootMembers *420*
set *420*
siblings *420*
sqrt *420*

subset 420
substring 421
tail 421
topCount 421
topPercent 421
topSum 421
trim 422
tuple 422
union 422
unique 422
upper 423
value 423
DB2 423
ascii 423
ceiling 423
char 423
chr 424
concat 424
date 424
day 424
dayname 424
dayofweek 424
dayofweek_iso 425
dayofyear 425
days 425
dec 425
decimal 425
difference 426
digits 426
double 426
event_mon_state 426
float 426
hex 426
hour 427
insert 427
integer 427
int 427
julian_day 427
lcase 428
left 428
length 428
locate 428
long_varchar 428
ltrim 428
microsecond 429
midnight_seconds 429
minute 429
month 429
monthname 429

quarter 430
radians 430
repeat 430
replace 430
right 430
round 430
rtrim 431
second 431
sign 431
smallint 431
soundex 431
space 431
substr 432
table_name 432
table_schema 432
time 432
timestamp 432
timestamp_iso 433
timestampdiff 433
to_char 433
translate 433
trunc 433
truncate 434
ucase 434
value 434
varchar 434
week 434
year 434
DB2 Cast 435
DB2 Math 436
DB2 Trigonometry 437
Informix 439
cardinality 439
char_length 439
concat 439
date 439
day 440
extend 440
hex 440
initcap 440
length 441
lpad 441
mdy 441
month 441
nvl 441
octet_length 441
replace 442
round 442
rpad 442

substr *442*
to_char *442*
to_date *443*
trunc *443*
weekday *443*
year *443*
Informix Math *443*
Informix Trigonometry *444*
MS Access *445*
ascii *445*
ceiling *445*
chr *445*
concat *445*
curdate *446*
curtime *446*
dayname *446*
dayofmonth *446*
dayofweek *446*
dayofyear *446*
hour *447*
instr *447*
lcase *447*
left *447*
length *447*
locate *448*
ltrim *448*
minute *448*
month *448*
monthname *448*
now *448*
position *449*
quarter *449*
right *449*
round *449*
rtrim *449*
sign *449*
space *450*
substr *450*
substring *450*
truncate *450*
ucase *450*
week *450*
year *451*
MS Access Cast *451*
MS Access Math *452*
MS Access Trigonometry *452*
Oracle *453*
add_months *453*
ascii *453*

ceil 453
char_length 453
chr 454
concat 454
decode 454
dump 454
greatest 454
initcap 454
instr 455
instrb 455
least 455
length 455
lengthb 455
lpad 455
ltrim 456
months_between 456
new_time 456
next_day 456
nls_initcap 456
nls_lower 457
nls_upper 457
nvl 457
replace 457
round 457
rpad 458
rtrim 458
sign 458
soundex 458
substr 458
substrb 459
{sysdate} 459
to_char 459
to_date 459
to_number 459
translate 459
trunc 460
trunc 460
{user} 460
vsize 460
Oracle Math 460
Oracle Trigonometry 460
Red Brick 462
ceil 462
concat 462
{current_user} 462
date 462
dateadd 463
datediff 463
datename 463

dec *463*
decimal *463*
decode *464*
float *464*
ifnull *464*
int *464*
integer *464*
length *465*
lengthb *465*
ltrim *465*
nullif *465*
positionb *465*
real *466*
round *466*
rtrim *466*
sign *466*
string *466*
substr *467*
substrb *467*
time *467*
timestamp *467*
timestamp *467*
to_char *468*
SQL Server *468*
ascii *468*
char *468*
charindex *468*
{current_user} *468*
datalength *469*
dateadd *469*
datediff *469*
datename *469*
datepart *469*
day *470*
difference *470*
getdate *470*
left *470*
ltrim *470*
month *471*
patindex *471*
replace *471*
replicate *471*
reverse *471*
right *472*
round *472*
rtrim *472*
sign *472*
soundex *472*
space *472*

str 473
stuff 473
year 473
SQL Server Cast 473
SQL Server Math 474
SQL Server Trigonometry 475
Teradata 476
account 476
add_months 476
bytes 477
case_n 477
char2hexint 477
characters 477
database 477
date 478
format 478
index 478
log 478
nullif 478
nullifzero 478
profile 479
random 479
role 479
session 479
soundex 479
substr 479
time 480
type 480
user 480
vargraphic 480
zeroifnull 480
Teradata Trigonometry 480
SAP BW 483
SAP BW Trigonometry 483
SAP BW Math 484
Sybase 484
ascii 484
char 485
charindex 485
datalength 485
dateadd 485
datediff 485
datename 486
datepart 486
day 486
difference 486
getdate 487
left 487
ltrim 487

month 487
patindex 487
rand 487
replicate 488
reverse 488
right 488
round 488
rtrim 488
soundex 488
space 489
str 489
stuff 489
substring 489
to_unichar 489
uhighsurr 490
ulowsurr 490
uscalar 490
year 490
Sybase Math 491
Sybase Trigonometry 491
Report Functions 493
_add_days 493
_add_months 493
_add_years 493
_age 493
_day_of_week 494
_day_of_year 494
_days_between 494
_days_to_end_of_month 494
_first_of_month 494
_last_of_month 495
_make_timestamp 495
_months_between 495
_week_of_year 495
_years_between 495
_ymdint_between 496
abs 496
AsOfDate 496
AsOfTime 496
BurstKey 496
BurstRecipients 496
ceiling 497
CellValue 497
character_length 497
ColumnNumber 497
CubeCreatedOn 497
CubeCurrentPeriod 497
CubeDataUpdatedOn 498
CubeDefaultMeasure 498

CubeDescription *498*
CubeIsOptimized *498*
CubeName *498*
CubeSchemaUpdatedOn *498*
exp *499*
extract *499*
floor *499*
GetLocale *499*
HorizontalPageCount *499*
HorizontalPageNumber *499*
InScope *500*
IsBursting *500*
IsCrosstabColumnNodeMember *500*
IsCrosstabRowNodeMember *500*
IsFirstColumn *500*
IsInnerMostCrosstabColumnNodeMember *500*
IsInnerMostCrosstabRowNodeMember *501*
IsLastColumn *501*
IsLastInnerMostCrosstabColumnNodeMember *501*
IsLastInnerMostCrosstabRowNodeMember *501*
IsOuterMostCrosstabColumnNodeMember *501*
IsOuterMostCrosstabRowNodeMember *501*
IsPageCountAvailable *501*
ln *502*
Locale *502*
lower *502*
mapNumberToLetter *502*
mod *502*
ModelPath *502*
Now *503*
nullif *503*
octet_length *503*
PageCount *503*
PageName *503*
PageNumber *503*
ParamCount *504*
ParamDisplayValue *504*
ParamName *504*
ParamNames *504*
ParamValue *504*
position *504*
power *505*
ReportAuthorLocale *505*
ReportCreateDate *505*
ReportDate *505*
ReportDescription *505*
ReportID *505*
ReportLocale *506*
ReportName *506*

ReportOption 506
ReportOutput 506
ReportPath 506
ReportProductLocale 506
ReportSaveDate 507
round 507
RowNumber 507
ServerLocale 507
ServerName 507
sqrt 507
substring 508
TOCHeadingCount 508
Today 508
trim 508
upper 508
URLEncode 508
Data Type Casting Functions 509

Appendix F: Producing Reports in Microsoft Excel Format 515

Unable to Load Images from the Cognos 8 Content Store in a Report 515
A Blank Worksheet is Opened 515
A Warning Message Appears When Excel Opens a Cognos 8 Report 515
Spreadsheet Content Not Saved for Reports Saved in XLS Format 516
Unable to Load Excel Report in Netscape 7.01 516
Unable to Nest Labels in Charts 516
Data Series Are Truncated 516
Colors in Excel Different From Those in HTML or PDF 516
Repeating Pie Charts 516
Unable to Skip Discrete Axis Labels 517
Unsupported Cognos 8 Formatting 517
Cell Contains Series of # 518
Reports with More Than 256 Columns Cannot Be Rendered in Excel 518
Table and Column Width 518
Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions 518
Number Format Becomes Currency Format in Japanese Excel 519
Report Shows Data in Wrong Columns 519
Unable to Access a Report on a Remote Server 519
Drill-through Reports Are Not Supported 519
Map Reports Are Not Supported 519
Unsupported Excel Formatting 520
Hyperlink Buttons Are Not Supported 520
Unable to View Reports in Excel Format Sent as Email Attachments 520
Unsupported Chart Properties in Excel 520
Unsupported Chart Types in Excel 521

Appendix G: Report Studio Object and Property Reference 523

Report Studio Objects 523
3-D Area 523
3-D Bar 523
3-D Combination Chart 524

3-D Line *524*
3-D Scatter Chart *524*
Angular Axis *524*
Angular Measure *525*
Area *525*
As of Time Expression *525*
Axis Title *525*
Bar *525*
Baseline *526*
Baseline *526*
Baseline *526*
Baseline *526*
Block *527*
Bookmark *527*
Bubble Chart *527*
Bubble Measure *527*
Calculated Measure *527*
Calculated Member *528*
Calculated Member *528*
Caption *528*
Chart Body *528*
Chart Footer *528*
Chart Node Member *529*
Chart Subtitle *529*
Chart Text Item *529*
Chart Title *529*
Class *529*
Combination Chart *530*
Component Override *530*
Conditional Block *530*
Conditional Block *530*
Context Item Text *531*
Crosstab *531*
Crosstab Columns *531*
Crosstab Columns *531*
Crosstab Corner *532*
Crosstab Fact Cells *532*
Crosstab Intersection *532*
Crosstab Member Fact Cells *533*
Crosstab Node Member *533*
Crosstab Rows *533*
Crosstab Rows *534*
Crosstab Space *534*
Cumulation Line *534*
Cumulation Line Axis *534*
Cumulation Line Label *534*
Data Item *535*
Date *535*
Date & Time Prompt *535*

Date Prompt *535*
Default Measure *535*
Detail Filter *536*
Dimension *536*
Display Layer *536*
Explicit Member Set *536*
Fact *536*
Field Set *537*
Gauge Chart *537*
Gauge Labels *537*
Gauge Numerical Axis *537*
Generated Prompt *537*
Hierarchy Set *538*
HTML Item *538*
Hyperlink *538*
Hyperlink Button *538*
Image *539*
Intersection (Tuple) *539*
Interval Prompt *539*
Join *539*
Key *539*
Layout Component Reference *540*
Legend *540*
Legend Title *540*
Level *540*
Level Hierarchy *541*
Level Set *541*
Line *541*
List *541*
List Cell *541*
List Column *542*
List Column Body *542*
List Columns *542*
List Columns *542*
List Columns Body Style *543*
List Columns Title Style *543*
List Column Title *543*
List Footer *543*
List Header *544*
List Page Footer *544*
List Page Header *544*
List Row *544*
List Row Cells Style *544*
Map *545*
Map Location *545*
Map Location *545*
Map Refinement Location *545*
Map Refinement Location *545*
Marker *546*

Marker *546*
MDX *546*
Measure *546*
Member *546*
Member Children Set *547*
Member Hierarchy *547*
Member Property *547*
Member Property *547*
Member Set *547*
Metrics Range Chart *548*
Metric Studio Diagram *548*
Named Set *548*
Note *548*
Note Content *548*
Numerical Axis *549*
Numerical Axis *549*
Ordinal Axis *549*
Page *549*
Page Body *549*
Page Break Text *550*
Page Footer *550*
Page Header *550*
Page Number *550*
Page Set *550*
Pareto Chart *551*
Pie Chart *551*
Pie Labels *551*
Point Color Legend Title *552*
Point Layer *552*
Point Measure *552*
Point Size Legend Title *552*
Point Size Measure *552*
Polar Chart *553*
Progressive Chart *553*
Prompt Button *553*
Query *554*
Query Operation *554*
Query Reference *554*
Radar Chart *554*
Radial Axis *554*
Radial Measure *555*
Region Color Legend Title *555*
Region Layer *555*
Region Measure *555*
Regression Line *555*
Repeater *556*
Repeater Table *556*
Repeater Table Cell *556*
Rich Text Item *556*

Row Number 557
Scatter Chart 557
Select & Search Prompt 557
Set Expression 557
Singleton 557
Slicer Member Set 558
SQL 558
Summary Filter 558
Table 558
Table Cell 558
Table of Contents 559
Table of Contents Entry 559
Table Row 559
Target Measure 559
Text Box Prompt 559
Text Item 560
Time 560
Time Prompt 560
Tolerance Measure 560
Total Column 561
Total Column Label 561
Tree Prompt 561
Value Prompt 561
Variable 561
X Axis 562
X Axis 562
X Axis Measure 562
Y1 Axis 562
Y2 Axis 562
Y Axis 563
Y Axis 563
Y Axis Measure 563
Z Axis 563
Z Axis 563
Z Axis Measure 564
Report Studio Properties 564
3-D Viewing Angle 564
Absolute Position 564
Across 564
Aggregate Function 565
Allow 45° Rotation 565
Allow 90° Rotation 565
Allow Skip 565
Allow Stagger 565
Angular Axis 566
Application 566
Apply Single Class 566
Auto Group & Summarize 566
Auto-Sort 567

Auto-Submit *567*
Auto Truncation *567*
Avoid Division by Zero *567*
Avoid Label Collision *567*
Axis Assignment *567*
Axis Assignment *568*
Axis Labels *568*
Axis Line *568*
Axis Title *568*
Background Color *568*
Background Image *569*
Bar, Line, or Area Index *569*
Baselines *569*
Block Variable *569*
Border *570*
Border Color *570*
Borders *570*
Bottom Position (px) *570*
Bottom Position (px) *571*
Box Type *571*
Box Type *571*
Calculation Intersection *571*
Calendar Type *572*
Caption *572*
Caption *572*
Cardinality *572*
Cascade Source *572*
Case Insensitive *572*
Catalog *573*
Chart Orientation *573*
Chart Type *573*
Chart Type *573*
Classes *573*
Clock Mode *574*
Color Legend Title *574*
Color Legend Title *574*
Column Coordinate *574*
Column Titles *574*
Component Reference *575*
Component Reference *575*
Conditional Palette *575*
Conditional Styles *575*
Connecting Lines *576*
Contents Height *576*
Cross Product Allowed *576*
Cumulation Axis *576*
Cumulation Label *576*
Cumulative Line *577*
Current Block *577*

Custom Label *577*
Data Format *577*
Data Item *577*
Data Item *578*
Data Item Label *578*
Data Item Label *578*
Data Item Label *578*
Data Item Label *578*
Data Item Value *579*
Data Item Value *579*
Data Item Value *579*
Data Item Value *579*
Data Item Value *580*
Data Item Value *580*
Data Language *580*
Data Source *580*
Default Measure *580*
Default Selections *580*
Default Title *581*
Define Contents *581*
Define Member Sets *581*
Depth *581*
Description *581*
Detail *581*
Diagram Identifier *582*
Dial Outline Color *582*
Dictionary *582*
Dimension *582*
Display After Overall Header *582*
Display Frequency *582*
Display Milliseconds *583*
Display Seconds *583*
Display Value *583*
Down *583*
Drill-Through Definitions *583*
Duplicates *584*
Embed *584*
Execution Method *584*
Execution Optimization *584*
Expand Features *584*
Expression *585*
Expression *585*
Expression *585*
Expression *585*
Face Color *585*
Fact Cells Precedence *586*
Fill Effects *586*
First Column Color *586*
First Date *586*

First Label Index *586*
Floating *587*
Font *587*
Font Auto-Sizing *587*
Footer *587*
Foreground Color *588*
Gauge Labels *588*
Gauge Palette *588*
Generated SQL/MDX *588*
Gridlines *589*
Grouping & Sorting *589*
Grouping Type *589*
Group Span *589*
Has Fact Cells *589*
Heading Level *589*
Height (px) *590*
Hide Adornments *590*
Hide Text *590*
Hierarchy *590*
Hole Size (%) *590*
Horizontal Alignment *591*
HTML *591*
HTML Source Variable *591*
Ignore Data with No Features *591*
Include Zero For Auto Scale *592*
Join Relationships *592*
Label *592*
Label *592*
Label *592*
Label *592*
Label *593*
Label *593*
Label Control *593*
Labels *593*
Labels *593*
Last Date *594*
Left Position (px) *594*
Left Position (px) *594*
Legend *594*
Legend Label *594*
Legend Title *594*
Level *595*
Level Indentation *595*
Line *595*
Line Style *595*
Line Styles *595*
Line Weight (pt) *596*
Map & Layers *596*
Map Drills *596*

Margin *596*
Marker Color *596*
Marker Label *596*
Markers *597*
Marker Text Location *597*
Master Detail Relationships *597*
Maximum Characters *597*
Maximum Execution Time *597*
Maximum Hotspots *598*
Maximum Rows Retrieved *598*
Maximum Size (pt) *598*
Maximum Tables *598*
Maximum Text Blob Characters *599*
Maximum Truncation Characters *599*
Maximum Value *599*
MDX *599*
Member *599*
Member Offset (%) *600*
Members *600*
Minimum Size (pt) *600*
Minimum Value *600*
Minor Gridlines *600*
Multi-Line *601*
Multi-Select *601*
Name *601*
Name *601*
Name *602*
Name *602*
Name *602*
Name *602*
Negative Column Color *602*
No Data Contents *603*
No Data Features Size (pt) *603*
Node Coordinate *603*
Note Border *603*
Notes *603*
Number of Regression Lines *604*
Numbers Only *604*
Number Style *604*
Numerical Axis *604*
Numerical Axis *604*
Numeric Value *604*
Numeric Value *604*
Ordinal Axis *605*
Outer Join Allowed *605*
Override Dimension Info *605*
Overrides *605*
Padding *605*
Pagination *606*

Palette *606*
Palette *606*
Parameter *606*
PDF Page Setup *606*
Percentile *607*
Percent of Axis *607*
Performance Pattern *607*
Pie Labels *607*
Point Color *607*
Point Shape *607*
Point Shape *608*
Point Size (pt) *608*
Polynomial Exponent *608*
Position *608*
Position type *608*
Position type *608*
Position type *609*
Positive Column Color *609*
Pre-populate *609*
Pre-populate Levels *609*
Pre-Sort *609*
Processing *610*
Progressive Axis *610*
Projection List *610*
Properties *610*
Property *610*
Push To Bottom *611*
Query *611*
Radar Type *611*
Radial Axis *611*
Range *611*
Range Label *612*
Regression Line *612*
Regression Type *612*
Relative Alignment *612*
Render Fact Cells *612*
Render Page when Empty *613*
Render Variable *613*
Repeater Direction *613*
Report Expression *613*
Report Expression *613*
Report Expression *614*
Report Expression *614*
Report Expression *614*
Report Expression *614*
Report Expression *614*
Report Expression *615*
Report Expression *615*
Required *615*

Right Position (px) *615*
Rollup Aggregate Function *615*
Rollup Processing *616*
Root Members Only *616*
Rotate Values *616*
Row Coordinate *616*
Rows Per Page *617*
Scale *617*
Scale Interval *617*
Scope *617*
Select UI *617*
Separator *617*
Series Color *618*
Set Operation *618*
Set Sorting *618*
Show Caption *618*
Show Data Range in Legend *618*
Show Features with No Data *619*
Show Feelers *619*
Show Legend Values *619*
Size & Overflow *619*
Size Legend Title *620*
Solve Order *620*
Sorting *620*
Source Type *620*
Source Type *620*
Source Type *621*
Source Type *621*
Spacing & Breaking *621*
Spider Effects *621*
SQL *621*
SQL join syntax *622*
SQL Syntax *622*
Standard Deviations *622*
Static Choices *622*
Style Variable *622*
Subtitle *623*
Suppress *623*
Table of Contents *623*
Table of Contents Name *623*
Table Properties *624*
Target Color *624*
Target Marker *624*
Target Marker Border Color *624*
Target Marker Position *624*
Target Range (%) *624*
Text *625*
Text Flow & Justification *625*
Text Source Variable *625*

Title 625
Tolerance Color 626
Tolerance Label 626
Tooltips 626
Top Position (px) 626
Total Column 626
Total Column Color 626
Truncation 627
Truncation Text 627
Type 627
Type 627
Type 627
Upper Range Skew (%) 628
URL 628
URL Source Variable 628
Usage 628
Use 1.x Behavior 628
Use Detail Value on Page 629
Use for Parameter Info 629
Use Local Cache 629
Use Same Range For All Instances 629
Use SQL With Clause 629
Use Thousands Separator 630
Use Value 630
Use Value 630
Value Location 630
Value Representation 630
Values 630
Values 631
Values 631
Value Type 631
Vertical Alignment 631
Visible 631
Visual Angle 632
White Space 632
Width (px) 632
X Axis 632
Y1 Axis 633
Y2 Axis 633
Y2 Axis Position 633
Y Axis 633
Data Formatting Properties 633
"Not Applicable" Characters 633
Any Error Characters 633
Calendar Type 634
Clock 634
Currency 634
Currency Display 634
Currency Symbol 634

Currency Symbol Position *634*
Date Ordering *634*
Date Separator *634*
Date Style *635*
Decimal Separator *635*
Display AM / PM Symbols *635*
Display As Exponent *635*
Display Days *635*
Display Eras *635*
Display Hours *635*
Display Milliseconds *635*
Display Minutes *635*
Display Months *636*
Display Months *636*
Display Seconds *636*
Display Time Zone *636*
Display Weekdays *636*
Display Years *636*
Display Years *636*
Divide By Zero Characters *636*
Exponent Symbol *636*
Group Size (digits) *637*
International Currency Symbol *637*
Mantissa (digits) *637*
Maximum No. of Digits *637*
Minimum No. of Digits *637*
Missing Value Characters *637*
Negative Pattern *637*
Negative Sign Position *637*
Negative Sign Symbol *638*
No. of Decimal Places *638*
Numeric Overflow Characters *638*
Padding Character *638*
Pattern *638*
Percentage Symbol *638*
Percent Scale (integer) *638*
Scale *638*
Secondary Group Size (digits) *639*
Security Error Characters *639*
Thousands Separator *639*
Time Separator *639*
Time Style *639*
Time Unit *639*
Use Thousands Separator *639*
Zero Value Characters *639*

Appendix H: Using Patterns to Format Data *641*

Pattern Guidelines *641*
Date and Time Symbols *642*

Decimal Format Symbols *651*

Appendix I: Drilling Through from Cognos Series 7 to Cognos 8 *653*

Setting Up Drill-through Access from PowerPlay Web *653*

Setting Up Drill-through Access from Cognos Visualizer *653*

Create and Test the Target Report *654*

Glossary *655*

Index *661*

Introduction

This document is intended for use with Report Studio. It provides step-by-step procedures and background information to help you create standard and complex reports. Report Studio is a Web product for creating reports that analyze corporate data according to specific information needs.

Audience

To use this guide, you should have

- knowledge of your business requirements
- experience using a Web browser, developing Web applications, and writing reports
- knowledge of databases and data modeling concepts

Related Documentation

Our documentation includes user guides, getting started guides, new features guides, readmes, and other materials to meet the needs of our varied audience. The following documents contain related information and may be referred to in this document.

Note: For online users of this document, a Web page such as **The page cannot be found** may appear when clicking individual links in the following table. Documents are made available for your particular installation and translation configuration. If a link is unavailable, you can access the document on the Cognos Global Customer Services Web site (http://support.cognos.com). Logon credentials are available either from your administrator or by request from support.america@cognos.com.

Document	Description
Cognos 8 Getting Started	Teaching new users how to use Cognos 8
Cognos Connection User Guide	Using Cognos Connection to publish, find, manage, organize, and view Cognos content, such as scorecards, reports, analyses, and agents
Cognos 8 Administration and Security Guide	Managing servers, security, reports, and portal services; setting up Cognos samples; troubleshooting; and customizing Cognos 8
Analysis Studio User Guide	Exploring, analyzing, and comparing dimensional data
Framework Manager User Guide	Creating and publishing models using Framework Manager

Document	Description
Metric Studio User Guide	Authoring scorecard applications and monitoring the metrics within them
Query Studio User Guide	Creating self-service business intelligence reports
Map Manager Installation and User Guide	Installing and using Map Manager to import and manage maps that are used in map reports

Finding Information

To find the most current product documentation, including all localized documentation, access the Cognos Global Customer Services Web site (http://support.cognos.com). Click the **Documentation** link to access documentation guides. Click the **Knowledge Base** link to access all documentation, technical papers, and multimedia materials.

Product documentation is available in online help from the **Help** menu or button in Cognos products. You can also download documentation in PDF format from the Cognos Global Customer Services Web site.

You can also read PDF versions of the product readme files and installation guides directly from Cognos product CDs.

Using Quick Tours

Quick tours are short online tutorials that illustrate key features in Cognos product components. To view a quick tour, start Cognos Connection and click the **Quick Tour** link in the lower-right corner of the Welcome page.

Getting Help

For more information about using this product or for technical assistance, visit the Cognos Global Customer Services Web site (http://support.cognos.com). This site provides product information, services, user forums, and a knowledge base of documentation and multimedia materials. To create a case, contact a support person, or to provide feedback, click the **Contact Us** link. For information about education and training, click the **Training** link.

Printing Copyright Material

You can print selected pages, a section, or the whole book. Cognos grants you a non-exclusive, non-transferable license to use, copy, and reproduce the copyright materials, in printed or electronic format, solely for the purpose of operating, maintaining, and providing internal training on Cognos software.

Chapter 1: What's New?

This section contains a list of new, changed, and deprecated features for this release. It also contains a cumulative list of similar information for previous releases. It will help you plan your upgrade and application deployment strategies and the training requirements for your users.

For information about upgrading, see the *Installation and Configuration Guide* for your product.

For information about new features for this release, see the *New Features Guide*.

To review an up-to-date list of environments supported by Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit the Cognos support Web site (http://support.cognos.com).

For information about upgrading, see the Installation and Configuration Guide for your product.

For an overview of new features for this release, see the *New Features Guide*.

New Features in Version 8.3

Listed below are new features since the last release. Links to directly-related topics are included.

Professional and Express Authoring Modes

Report Studio now accommodates two distinct types of users:

- the professional report author
- the report author

The Professional authoring mode is designed for report authors who require access to the full range of Report Studio functionality. This is the traditional Report Studio profile and the only profile available until this release.

The Express authoring mode is a subset of the traditional Report Studio user interface and is aimed at creating financial crosstab reports using dimensional data sources. Financial authoring requires many but not all of the features that already exist in Report Studio, combined with a more intuitive user experience and interaction with live data.

For more information, see "Authoring Modes" (p. 61).

Enhanced User Interface When Working With Dimensional Data Sources

Report Studio now offers several enhancements when working with a dimensional data source. You can now

- view a members-oriented source tree in the **Insertable Objects** pane and customize the tree to show the information you want (p. 66).

- create sets of members by selecting multiple members from within the same hierarchy in the source tree (p. 169).
- select whether to insert only members or only their children or both into your report (p. 169)
- create sectioned page breaks by dropping members or sets in the **Page layers** area (p. 181).
- create slicer filters by dropping members or sets in the **Context filter** area (p. 174).

Enhanced Conditional Styles

Conditional styles are now easier to use, due to a new streamlined user interface and more options.

You can now apply multiple conditional styles in reports. For example, you can apply one style for data formats on specific objects, and apply a second style for conditional highlighting throughout the report.

For more information, see "Highlight Data Using Conditional Styles" (p. 254).

Extended Crosstab Indentation

Crosstab indentation is extended with new properties to support both outdenting and relative indenting. Relative indenting adds an indent based on the levels as they appear in the report rather than in the cube data source. You can now avoid double-indenting, by which members are taken from both the parent and grandchild levels.

For more information, see "Indent Data" (p. 91).

Customizable Data Tree

You can now customize the source tree in the **Insertable Objects** pane. You can select options for a metadata view with dimensions, hierarchies, levels, and an optional members folder; or you can select a member-oriented tree view. You can also select a combination of these views. For more information, see "Customize Your Source Tree" (p. 66).

Table of Contents in PDF Reports

You can now add one or more book-like table of contents in your PDF report to summarize all or part of the content of the report. The table of contents provides page numbers and dynamic links to the content.

For more information, see "Create a Table of Contents" (p. 140).

No Data Handling

When a query returns no data, you can now choose to provide a message or to remove the data item from the report. The new **No Data Contents** property was added to many data containers such as lists, crosstabs, and charts. For more information, see "Specify What Appears for a Data Container that Contains No Data" (p. 199).

You can also choose to not render the entire report page if all the data containers on the page do not contain any data. For more information, see "Specify Not to Render a Page If It Does not Contain Data" (p. 70).

Simplified Report Style

You can now use a Report Studio style class named **Simplified styles** to remove the default styles defined in the GlobalReportStyles.css file. This class can be used in any report to present a style more appropriate to the financial crosstab style of report. This new class is used in the financial report template.

For more information, see "Create and Modify Classes" (p. 164).

Enhanced Report Expressions

A series of new functions was added in the expression editor to make report expressions more powerful. The new functions give you the flexibility to create report expressions for reporting and conditional processing.

The new functions include a new mathematical function named mod(), time-based functions, and conversion functions. Of interest to PowerCube users are the cube variables that can now be added to report output.

The double2string( ) function can be used in data type conversions.

The following functions return the named PowerCube properties:

- CubeName( )
- CubeDescription( )
- CubePath( )
- CubeCreatedOn( )
- CubeDataUpdatedOn( )
- CubeSchemaUpdatedOn( )
- CubeIsOptimized( )
- CubeDefaultMeasure( )
- CubeCurrentPeriod( )
- CellValue( )

The following functions are used to determine where the current cell is within a data frame, such as a list or crosstab:

- GetColumnNumber( )
- IsFirstColumn( )
- IsLastColumn( )

- GetColumnNumber( )
- GetRowNumber( )
- IsColumnNodeMember( )
- IsRowNodeMember( )
- IsInnerMostColumnNodeMember( )
- IsOuterMostColumnNodeMember( )
- IsInnerMostRowNodeMember( )
- IsOuterMostRowNodeMember( )
- IsLastInnerMostColumnNodeMember( )
- IsLastInnerMostRowNodeMember( )
- IsFirstColumn( )
- IsLastColumn( )
- IsFirstRow( )
- IsLastRow( )
- GetTableRowNumber( )
- GetTableColumnNumber( )
- InScope( )

For more information, see "Using the Expression Editor" (p. 357).

Singleton Results in Layout

You can now insert single data items anywhere in your report using the new singleton object.

For more information, see "Insert a Single Data Item" (p. 196).

PDF Horizontal Pagination

Reports can now span two or more page widths in a PDF. This is useful when you want wide data in lists and crosstabs to show in its original size. The fit-to-page option from previous releases is still available. You decide for each list or crosstab which items span across pages and which items shrink to fit a single page.

Several options for horizontal page numbering are provided. For more information, see "Control Page Breaks and Page Numbering" (p. 158) and "Insert Page Numbers" (p. 157).

Page-by-Page PDF Page Layout

You can now set the page orientation (portrait or landscape) and paper size independently on each page of PDF reports. This is useful when some pages within the same report present information that is better suited to a different layout.

The page orientation and size are set as properties of the page. For more information, see "Set PDF Page Options" (p. 73).

Zooming on Chart Region

You can now zoom into any area of a map report. Because the area of interest is expanded to fit the full report frame, it is no longer necessary to create a separate map in MapInfo for each area of interest or the corresponding drill-through and report definitions.

A single map with multiple layers can be used as the target of authored reports that use a drill-through or a prompt to define the area of interest. The new **Expand Features** property indicates whether the map should be expanded to focus on the filtered features.

For example, a single map of Europe first shows revenue by country and then, when drilling on a country, expands the selected country to fit the report frame.

For more information, see "Expand Features" (p. 584).

Other New Features

Other new features include the following:

- A new style toolbar allows you to easily apply styles, such as fonts, colors, alignment, borders, and data formats.
- You can now copy and reuse the style information from one object to another (p. 139).
- You can now save your report output to a file system, such as a local area network "Save a Report" (p. 66).
- A new financial report template is now available with default formatting for financial reports (p. 63).
- The **Select with Search** and **Value** prompts now let you browse through large numbers of prompt values (p. 201).
- You can now perform a search to find objects with specific characteristics in your report (p. 57).
- You can now specify a default measure for map, point, scatter, bubble, and polar charts (p. 198).
- Excel 2007 native spreadsheets are now supported as a report format, in addition to the existing Microsoft Excel HTML formats "Producing a Report in Excel Format" (p. 72)"Run a Report" (p. 66).

Changed Features in Version 8.3

Listed below are changes to features since the last release. Links to directly-related topics are included.

Product Behavior After Upgrade

When you upgrade from Cognos 8 BI version 8.2 to version 8.3, some features in Cognos 8 may behave differently after the upgrade. When you upgrade reports, for example, changes in behavior may cause validation errors. Documentation is available about the behavior changes. This documentation includes examples of the changed behavior and solutions for issues that may occur during the upgrade. For more information, see *Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior* on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Special Cell Values Shown Differently

The default formatting for special cell values in reports originally created in Analysis Studio has changed from 8.2 to 8.3. Special cell values include nulls and missing values, division by zero values, values denied by security, N/A, overflow, or other error values.

You can change the characters for special cell values. For more information, see "Format Data" (p. 193).

Cell Values	Cognos 8.3	Cognos 8.2 and Earlier
Nulls and missing values	(blank)	(blank)
Overflow or underflow	--	#!Overflow
Division by zero	/0	/0
Security (for example, insufficient access rights)	#!Security	#!Security
N/A	--	(blank)
Error	--	#!Error
Casting error	--	#!Casting Error

Slicer Filters Now Affect All Edge Values

Slicer filters now apply to all edge expressions, as they do to cell values. For more information, see *Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior* on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Default Calculation Position

The default position for calculations in Analysis Studio has changed from 8.2 to 8.3. In previous releases, calculations were positioned at the end of a set. In a crosstab that contains many items, the calculation may not be visible on the screen. The default position of calculations is now after the last item used as an operand in the calculation.

The position of a calculation in a saved analysis created in Analysis Studio 8.2 may appear in a new position in the crosstab when viewed in Analysis Studio, Report Studio or Cognos Viewer in Cognos 8.3. This does not apply to reports originally created in Analysis Studio and then saved in Report Studio.

Changes to Default Data Formats for OLAP Data Sources

The default formats for calculated values have changed when you use the following OLAP data sources:

- Dimensionally modeled relational data sources
- Cognos PowerCube
- Cognos 8 Planning - Contributor
- Essbase
- SAP BW

The changes may affect the number of decimal places, group separators, or the rounding precision used in your reports. The following table lists some examples of the changes.

Data format or calculation	Description of change
Group separators	Large percentage results now include group separators (typically a comma at every three digits). For example, 1234.00% becomes 1,234.00%. If none of the values in a calculation or summary use group separators, the results also do not use group separators.
$0 format	Measures that were formatted with the Cognos format string $0 (possible with Essbase) now have decimal places, since this format string doesn't specify that decimals should be suppressed.

Data format or calculation	Description of change
Division	If the numerator is a currency with at least one decimal, and the denominator is a query constant, a number, or a percentage, the result now has as many decimals as the currency. In all other cases, the result of a division always has three decimals. If the denominator is a query constant, a number, or a percentage, the result now has t, he same format type as the numerator. In all other cases, the result of a division is a number.
Square root	If the operand is a currency with decimals, its decimal precision is now used for the result. Otherwise the result has three decimal places.
Rounding	The results now have as many decimals as the rounding precision.

You can override these default formats using Report Studio or Query Studio. For more information, see the Knowledge Base on the Cognos support Web site (http://support.cognos.com).

Summaries on Non-Additive Measures Give Incorrect Results for SAP BW

If your list or grouped list report includes a summary, such as a total, average, or variance, on a non-additive measure (such as the price of an item, or the quantity on hand in an inventory system) , error cells, which typically show two dashes (--), are now returned. However, when using an SAP BW data source, the first value is returned as the summary for the non-additive measure instead of two dashes.

In Query Studio, a meaningless number may now appear instead of two dashes.

To obtain correct results, in Report Studio, set the Suppress property for the query to **None**. This option may cause more empty rows to appear in the report. You can also remove the summaries from your report. The report will no longer include a footer with two dashes or empty cells as the summary.

FOR Aggregate is No Longer Context-dependent for OLAP

If a crosstab report against a relational data source uses a `for` clause in an aggregate expression, the results do not depend on whether the expression is projected on another edge.

In Cognos 8.1 and 8.2, in many cases for OLAP data sources, this was not true - the expression was calculated in the context of the opposite edge. As a result, some charts in Query Studio often did not match the corresponding list report.

In Cognos 8.3, the OLAP behavior is now consistent with the relational behavior. If you required context-dependence, use either the bare measure, or the member aggregate syntax (`within set` or `within detail`).

For more information, see *Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior* on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Rounding with Equality Filters for MSAS Cubes

In previous versions of Cognos 8, when using an MSAS 2000 or 2005 cube, adding an equality filter with a specific constant implicitly rounded the cells to the number of decimal places for the constant. This caused some inconsistencies when using the following operators: <, >, <=, >=, and <>.

For example, the following filter uses the not equal (< >) operator: [measures].[BigInt] <> 3.14159. It now returns the raw values 3.141586 and 3.141592, and no longer rounds the values to five decimal places.

For example, the following filter uses the = operator: [MEASURES].[BigInt] = 3.14159. In previous releases, the left side of the expression was rounded to 5 digits and raw values of 3.141586 and 3.141592 were returned. In this release, raw values of 3.141586 and 3.141592 are no longer returned, consistent with the behavior for <>.

Other data sources were always consistent and did not round values.

If you want to keep the old behavior, use the `round` function, or use ([MEASURES].[BigInt] < 3.141595 AND [MEASURES].[BigInt] >= 3.141585).

Essbase Calculations on Missing Facts

In Cognos 8 version8.2, arithmetic query calculations for Essbase data sources treated missing fact values as zero. In version 8.3, such calculations now have no value (null).

This new behavior is consistent with the behavior of calculated members in Essbase cubes, and with PowerCubes and SAP BW behavior. In addition, it improves the behavior of null suppression based on such calculations.

If you want to keep these values as zero, you can explicitly convert the null to zero with an expression such as if ([Discount Percentage] IS NULL) then (0) else ([Discount Percentage]).

For more information, see *Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior* on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Changes to the Data Tree for SAP BW

If you use an SAP BW data source, you will notice the following changes when working with the data tree:

- If you expand the member folder, you now see all the members.
- If you expand the member folder and the data source uses variables, you are no longer prompted to satisfy the variables.

- If you search in the member folder, you are no longer prompted to satisfy the data source variables.
- If you search for members, the search retrieves all the members that meet the search criteria despite the variable conditions.
- If your data source uses variables and the report includes a Cognos 8 prompt, you are now prompted to satisfy the variables.

Deprecated Features in Version 8.3

A deprecated feature is one that is being replaced by a newer version or a better implementation. Cognos intends to discontinue the use of the feature and provides recommendations for adapting to this change over multiple releases.

Listed below are deprecated features, including links to related topics.

Excel 2000 Format

The Microsoft Excel 2000 format for report outputs is deprecated in version 8.3. In a future release, Excel 2000 format will no longer be available.

Chapter 2: Report Studio

Report Studio is a Web-based tool that professional report authors use to build sophisticated, multiple-page, multiple-query reports against multiple databases. With Report Studio, you can create any report that your company requires, such as invoices, statements, and weekly sales and inventory reports.

Planning Reports

Before you use Report Studio to design and create a report, some information is required. For each report, answer the following questions:

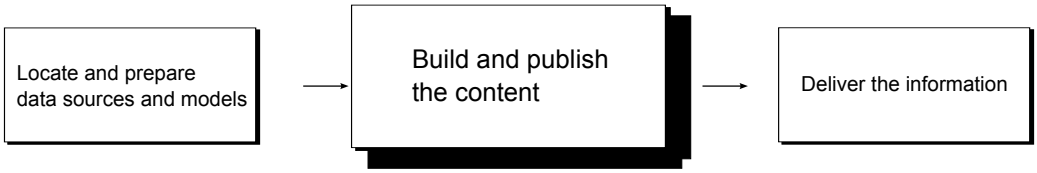
- Who is the target audience?
- What business question do you want to answer?
- Which type of report will best answer the business question?
- What are the data sources, and where are they?
- Which data items and calculations are needed?
- What are the security needs?
- How will the report be distributed?

Answering some of these questions involves working with other Cognos 8 components. For example, Framework Manager is the metadata tool used to model your enterprise data. Models created in Framework Manager provide the data items that you use in reports.

Building Cognos 8 Applications

The lifetime of a Cognos 8 business intelligence application can be months, or years. During that time, data may change, and new requirements appear, so authors must modify existing content and develop new content, and administrators must update models and data sources as the underlying data changes. For more information about using data sources, see the *Administration and Security Guide* and the Framework Manager *User Guide*.

In a working application, the technical and security infrastructure and the portal are in place, as well as processes for change management, data control, and so on. For information about the workflow associated with creating Cognos 8 content, see the *Architecture and Deployment Guide*. For more information, see the Cognos Solutions Implementation Methodology toolkit, which includes implementation roadmaps and supporting documents. The Cognos SIM toolkit is available on the Cognos Global Services Web site, http://support.cognos.com/en/consulting/csim/index.html.



- locate and prepare data sources and models

  Cognos 8 can report from a wide variety of data sources, both relational and dimensional. Database connections are created in the web administration interface, and are used for modeling, for authoring and for running the application.

  To use data for authoring and viewing, the business intelligence studios need a subset of a model of the metadata (called a package). The metadata may need extensive modeling in Framework Manager.

- build and publish the content

  Reports, scorecards, and so on are created in the business intelligence studios of Cognos 8. Which studio you use depends on the content, lifespan and audience of the report, and whether the data is modeled dimensionally or relationally. For example, self-service reporting and analysis are done through Query Studio and Analysis Studio, and scheduled reports are created in Report Studio. Report Studio reports and scorecards are usually prepared for a wider audience, published to Cognos Connection or another portal, and scheduled there for bursting, distribution, and so on. You can also use Report Studio to prepare templates for self-service reporting.

- deliver and view the information

  You deliver content from the Cognos portal or other supported portals, and view information that has been saved to portals, or delivered by other mechanisms. You can also run reports, analyses, scorecards, and so on from within the business intelligence studio in which they were created.

  For information about tuning and performance, see the *Administration and Security Guide* and the Cognos Global Services Web site, http://support.cognos.com.

Working in Report Studio

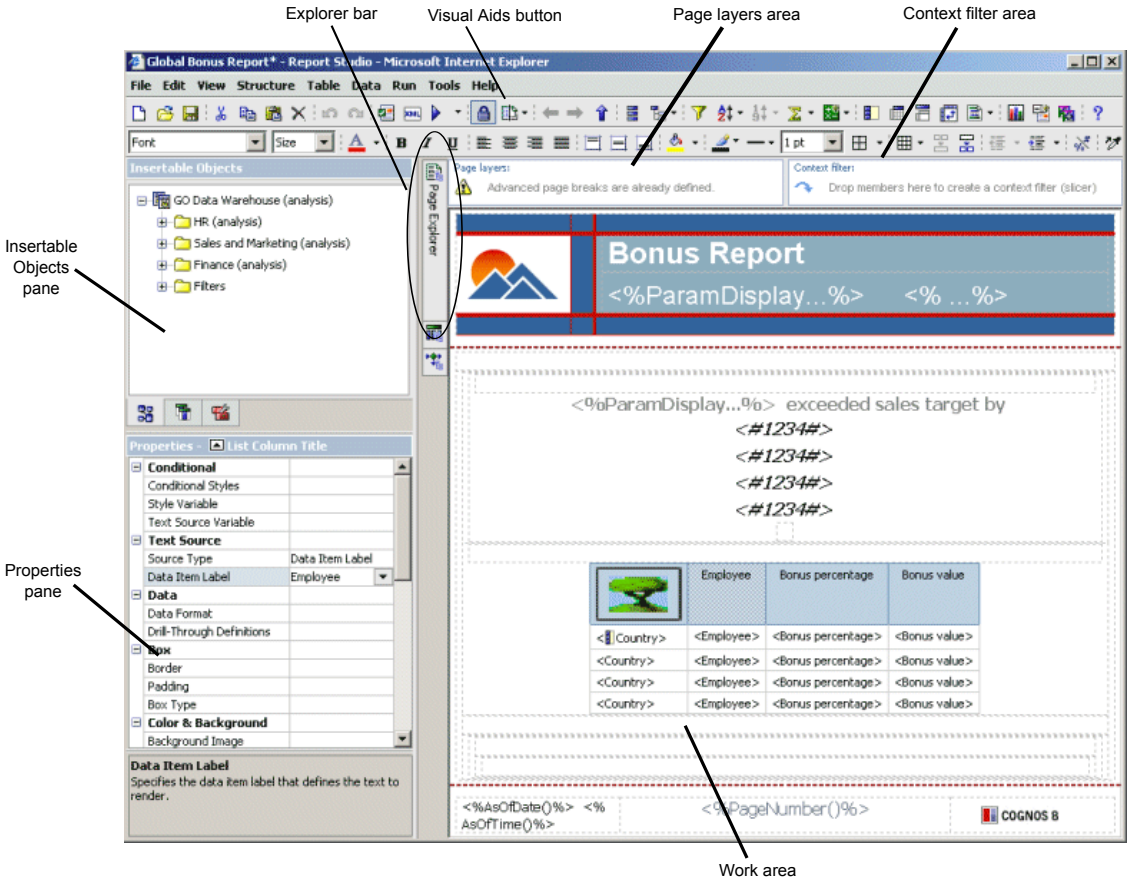
To create reports in Report Studio, you must become familiar with the Report Studio environment.

To work effectively in Report Studio, do the following:

- Familiarize yourself with the user interface.
- Learn about basic report structure, which includes the layout and queries.
- Learn how to work with report objects.
- Set the options you want.
- Verify Web browser settings.
- Learn about the different authoring modes.

The User Interface

The Report Studio user interface has two panes, an explorer bar, and a work area to help you create reports.



Insertable Objects Pane

The **Insertable Objects** pane contains objects that you can add to a report. You add objects to a report by dragging them to the work area.

The **Insertable Objects** pane contains these tabs:

- The **Source** tab contains items from the package selected for the report, such as data items and calculations.
- The **Data Items** tab describes the queries created in the report.
- The **Toolbox** tab contains a variety of objects that you can add to the report, such as text and graphics.

Properties Pane

The **Properties** pane lists the properties that you can set for an object in a report.

You can obtain additional information about a property by selecting it and clicking F1. For example, you can view the list of objects that uses each property.

When you specify a value for a property, press Enter, click another property, or save the report to ensure that the value is saved.

Tip: View a description of the currently selected property at the bottom of the pane. From the **View** menu, click **Property Descriptions**.

Explorer Bar

Use the following buttons on the Explorer bar to work with different parts of a report:

- Pause the pointer over the page explorer button to go to a specific report page or prompt page.

  Tip: You can also add a new report page or add a prompt page (p. 199), as well as create and modify classes (p. 164).

- Pause the pointer over the query explorer button to work with queries (p. 213).

  You use Query Explorer to create or modify queries and to perform complex tasks, such as defining union joins and writing SQL statements.

- Pause the pointer over the condition explorer button to work with variables.

  You use variables to define conditions (p. 254) in a report.

Page Layers Area

Use the **Page layers** area to create sections, or page breaks, in a report to show values for each member on a separate page. For example, you can drag **Northern Europe** sales territory from the **Insertable Objects** pane to the **Page layers** area. The report is broken into a separate page for each territory within northern Europe. Each page's context appears in the report header.

For more information, see (p. 181).

Context Filter Area

When working with dimensional data, use the **Context filter** area to filter your report to show values, or context, only for a specific data item. This technique is also known as a slicer filter. For example, you can drag **Sales Territory** from the **Insertable Objects** pane to the **Context filter** area. When you click a specific territory from the list, the values in the crosstab change to represent data for that territory.

For more information, see "Filter Values Using Context" (p. 174).

Work Area

The work area is where you design reports.

Visual Aids Button

The visual aids button provides the following options to help you when you are designing reports in the layout.

Option	Description
Show Boundary Lines	Shows all boundary lines.
Force Boundary Lines	Overrides all boundary line settings by replacing them with default (dotted line) lines. For example, if you specified a value in the **Border** property for an object, the border lines are replaced with dotted lines.
Show Repeating	Repeats objects when you insert them. For example, when you insert a data item in a crosstab, the data item appears in each row or in each column of the crosstab.
Show Page Header And Footer	Shows the page header and page footer.
Show Drag & Drop Padding	Shows drag-and-drop zone when the **Padding** property for an object is set to **0**. If the **Padding** property is set to a value that is greater than the minimum padding that Report Studio uses to show drag-and-drop zones, only the minimum padding is shown.
Show Hidden Objects	Shows objects for which the **Box Type** property was set to **None** or for which the **Visible** property was set to **No**.
Show Sorting	Shows the sorting icon for data items for which a sort order (p. 177) was specified.
Show Grouping	Shows the grouping icon for grouped data items (p. 80).
Show Source Type	Shows the icon for the source type of objects, such as layout calculation.
Show Data Item Type	Shows the icon for the type of data item, such as query item, member, or measure (p. 64).
Show Drill-through Definitions	Shows data items for which a drill-through definition (p. 273) was defined as hyperlinks.
Show Table of Contents Entries	Shows table of contents entries (p. 140) inserted in the report.
Show Bookmarks	Shows bookmarks (p. 141) inserted in the report.

Option	Description
Show Master Detail Relationships	Shows master detail relationships (p. 282) defined in the report. **Tip:** Pausing the pointer over the master detail relationship icon shows the relationship.
Show No Data Contents Tab Control	Shows tabs if the data container's **No Data Contents** property is set to **Yes** (p. 199).
Show Repeater and Singleton Containers	Shows repeater and singleton containers inserted in the report.

Basic Report Structure

All reports have two components, a layout component that defines the report appearance, and a query component that defines report data. Understanding these components will help you design effective reports.

Layout

A layout is a set of pages that defines the appearance and formatting of a report. When you design the layout of a report, you

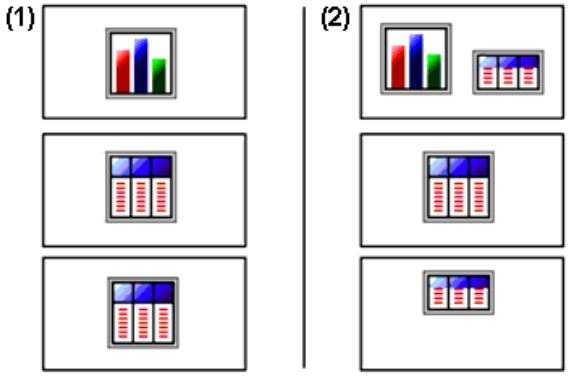
- present the data in a meaningful way by using lists, crosstabs, charts, and maps
- give the report the appearance you want by adding formatting, such as borders, color, images, and page numbers
- specify how the data flows from one page to the next

Pages

Pages are containers for the layout objects that you use to build a report. A page is made up of the following mandatory and optional components:

- page header (optional)
- page body (mandatory)
- page footer (optional)

When you run a report, the amount of data queried often exceeds one page. As a result, a page will repeat until all the data is shown. You have control over how data flows from one page to the next. For example, here are alternative representations of a report that contains a chart and a lengthy list.



1. The chart appears on its own page. The list begins on the next page and fills subsequent pages until all rows appear.
2. The chart and the first few rows of the list appear on the first page. The rest of the data in the list appears on the subsequent pages.

Objects

You add layout objects to a page when you create a report. Below are objects that you will use often when building reports in Report Studio:

- list

  Add a list to show data in rows and columns.

- crosstab

  Add a crosstab to show data in a grid, with dimensions along the rows and columns, and measures in the cells or intersection points.

- chart
- map
- repeater

  Add a repeater to show each instance of a certain column or data item in a separate frame.

- text
- block

  Add a block to hold text or other information. Blocks are often used to lay out horizontal bands of information.

- table

For information about other layout objects you can add to a report, see "Insert Other Objects" (p. 143).

Queries

Queries determine what data items appear in the report. Sometimes you want detail rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using aggregate functions and grouped columns, or must apply filters to show only the data you want.

Report Studio automatically creates the queries you need as you build reports. However, you can modify these queries or create your own custom queries to get the results you want.

For more information about queries, see "Working with Queries" (p. 213).

Working with Objects

You build reports by adding objects and manipulating them to obtain the results you want. To understand how to work with objects in Report Studio, you must be familiar with the following concepts:

- object types
- objects as containers
- locking and unlocking objects
- hierarchy of objects

For information about Report Studio objects, see "Report Studio Object and Property Reference" (p. 523).

Object Types

In Report Studio, layout objects are either inline or block. You can insert other objects on the same line as an inline object, but not on the same line as a block object. When you insert an object to the left or to the right of a block object, the object appears on the line above or below the block object respectively. Examples of inline objects include graphics and text items. Examples of block objects include any report type (list, crosstab, chart, map, or repeater) and tables.

Note: You can also use an object's floating (p. 162) property to define how other objects flow around the object. For example, you can specify how text flows around an image.

Objects as Containers

Objects, such as tables (p. 147), blocks (p. 143), and any report frame (p. 79), are containers in which you can insert other objects. For example, you can insert a list in one cell of a table and a chart in another.

Tip: You can also nest objects to create a sophisticated layout. For example, you can insert a table in a cell of another table.

Locking and Unlocking Objects

To manipulate the contents of some objects, you must first unlock the object. For example, you have a list that contains the column Product Name. You want to insert a graphic inside the Product Name column to show an image of each product. Unlocking the list allows you to insert the image object inside a list column.

Tip: From the **Structure** menu, click **Lock Page Objects** . Toggling this menu item locks and unlocks all layout objects in a report. However, this setting is not saved with the report.

Hierarchy of Objects

In Report Studio, objects are organized hierarchically. For example, a list contains list columns, and each list column contains a text item, which is the name of the inserted data item.

The hierarchy of objects is useful to remember when you apply formatting because formatting is applied to the child objects of the object. For example, you can specify that all list column titles in a list have red as the background color. The formatting is automatically applied to any new columns you add to the list, because the formatting is applied to the list and is therefore applied to the objects in the list. If you apply formatting to a specific object, it will override the same formatting specified for the parent object.

Find Objects in a Report

You can quickly locate specific objects in a report by using **Find**.

Steps

1. From the **Edit** menu, click **Find**.
2. Click **Find what** and choose the type of object that you want to find.
3. In the **Options** box, specify the search criteria that you want.

   For example, if you want to find the objects that reference a specific data item, you must specify the query that contains the data item and the data item name.
4. Click **Find Next**.

The first object that meets the search criteria is selected in the report. Continue clicking **Find Next** to show all other objects that meet the search criteria.

Note: In some cases, such as searching for a data item that is used to define a conditional style, Report Studio cannot directly select the object in the report. Instead, Report Studio selects the object that uses the object that you are searching for.

Set Options

Set the options you want when working in Report Studio.

View Options

Option	Description
Use Windows skin	Replaces the current appearance of the interface with the display scheme specified by the Windows settings.
Show startup dialog	Shows the **Welcome** dialog box at startup.
Reuse Cognos Viewer window	Reuses the same Cognos Viewer window when you rerun a report without first closing the window.

Option	Description
Resize Cognos Viewer window	Maximizes the Cognos Viewer window when you run a report.
Animate explorers	Animates the appearance of the **Page Explorer**, **Query Explorer**, and **Condition Explorer** (p. 52).
Window startup size	Specifies the size of the Report Studio window at startup.

Edit Options

Option	Description
Wrap text in editors	Automatically wraps text in all editors where you can define expressions.
Automatically populate values list	When building expressions in the expression editor, automatically shows values when browsing the data of a data item (p. 359).
Automatically validate expressions	Automatically validates expressions, such as filters, created in the expression editor (p. 357).
In-place edit	Enables the editing of text in place when double-clicking.
Allow drop replace in crosstab	When dropping an item from the **Insertable Objects** pane over an existing item in a crosstab, replaces the original item.
Double click insertion location	When working with dimensional data, specifies where the child members are inserted when you double-click the member data item.
Layout dimensions	Specifies the width and height of the area where you will create reports.

Report Options

Option	Description
Alias member unique names	When working with a dimensional data source, creates an alias when you add a member to the report or to an expression (p. 360).

Option	Description
Delete unreferenced query objects	Automatically deletes query objects linked to another object. For example, if you delete a list, the query linked to the list is deleted as well.
Delete unreferenced conditional styles	Automatically deletes conditional styles (p. 254) when the last data item that refers to the conditional style is also deleted.
Always create extended data items for new reports	Determines whether Report Studio creates extended data items or expression-based data items. Expression-based data items allow you to view or edit the expression by double-clicking the **Expression** property for the item. If you are working with a dimensional data source and this option is not selected, the appropriate dimensional data item is added when you insert items such as members, levels, or calculated members. For example, if you insert a level, a level set is created. This makes it easier to work with dimensional data items because Report Studio knows the data item type of the items that are in the report.
Limit on inserted members	When working with a dimensional data source, limits the number of child members that are inserted. For example, you specify 3 for this option and in the toolbar you specify the option to insert children when you drag a member to a data container (p. 169). You then drag the Camping Equipment member to the rows of a crosstab. What you see as rows are the child members Cooking Gear, Tents, and Sleeping Bags, and a row named Others (Camping Equipment) for the remaining child members of Camping Equipment.

Option	Description
Aggregation mode	When working with a dimensional data source, specifies the aggregation type you want Report Studio to use when aggregating values in crosstabs and charts. **Within detail** aggregates the visible details. **Within aggregate** aggregates the visible aggregates at the next lower level of detail. **Within set** aggregates the member sets. For more information about aggregating values in crosstabs and charts, see "Aggregating Values in Crosstabs and Charts" (p. 186).

Advanced Options

Option	Description
Use operating system clipboard	If Report Studio is running in Internet Explorer, uses the Windows clipboard instead of the internal application clipboard. **Tip:** If Report Studio is running in Firefox, the internal application clipboard is always used.
Allow local file access	Enables the opening and saving of reports on your computer (p. 235).
Map feature display limit	When working with maps, specifies the maximum number of features that can appear in a map.
Member display count limit (in source tree)	When working with dimensional data, specifies the maximum number of members that can appear in the **Insertable Objects** pane when performing a search (p. 171).

Steps

1. From the **Tools** menu, click **Options**.
2. On the **View**, **Edit**, **Report**, and **Advanced** tabs, set the options you want.
3. Click **OK**.

Web Browser Settings

Cognos 8 uses the default browser configurations provided by Microsoft, Netscape, and Firefox. For all browsers, you must ensure that settings are enabled for cookies and Java scripts. Additional required settings are specific to the browser.

Browser	Setting
Internet Explorer	Allow Cookies
	Active Scripting
	Allow META REFRESH
	Run ActiveX controls and plug-ins
	Script ActiveX controls marked safe for scripting
	Binary and Script Behaviours
	Allow programmatic clipboard access

Note: Report Studio uses the native Microsoft Internet Explorer XML support, which is a component of the browser. ActiveX support must be enabled because Microsoft implements XML using ActiveX. Cognos 8 does not provide or download ActiveX controls. Only the ActiveX controls that are installed as part of Internet Explorer are enabled through this configuration.

For more information about browser configuration and cookies used by Cognos 8, see the *Installation and Configuration Guide*.

Authoring Modes

To meet the needs of both regular report authors and financial report authors, Report Studio provides distinct custom user interfaces that contain reporting features relevant to these roles.

Access to each authoring mode is determined by the permissions you have to secured functions and features. For more information, see the *Administration and Security Guide*.

Professional Authoring Mode

The Professional authoring mode gives users access to the full range of Report Studio functionality. In this mode, you can create any report type, including charts, maps, lists, and repeaters, using any data source (relational or multi-dimensional). It contains a superset of the features available in the Express authoring mode. However, you cannot view live data.

Express Authoring Mode

The Express authoring mode provides a simplified and focused Report Studio interface. It is designed for non-technical users to create traditional financial and management statement reports. It only allows access to dimensionally-modeled data and uses a member-oriented data tree.

This authoring mode allows you to see live data and supports only crosstab reports. It contains a subset of the features available in the Professional authoring mode.

When you are in the Express authoring mode, if you open a report that was authored in the Professional authoring mode, you can see but cannot modify objects that can be inserted only in the Professional authoring mode, such as charts, maps, and lists.

Change Authoring Modes

If you have access to both the Professional and Express authoring modes, you can switch between the two from within Report Studio.

You may want to start authoring your report in the Express authoring mode to see live data and easily create your report layout. Then, you can switch to the Professional authoring mode to add more information, such as charts and multiple pages.

Step

- From the **View** menu, click **Authoring Mode** and select the one you want.

Creating a Report

When you create a report, you are actually creating a report specification. The report specification defines the queries and prompts that are used to retrieve data, as well as the layouts and styles used to present the data. For simplicity, the report specification is named the report.

Before creating a report, ensure that you have the planning information (p. 49) you need.

Creating a new report involves

- ❑ specifying the package
- ❑ choosing a report template
- ❑ adding data items
- ❑ saving the report
- ❑ running the report

You can then format the report (p. 133) and manipulate the data that will appear in the report (p. 167). For information about other tasks that are specific to a report type, see "Types of Reports" (p. 79).

Specify the Package

Specify the package that will provide items for the report.

The packages that you use to generate reports are based on models that are created in the modeling tool, Framework Manager. A model is a set of related objects, such as query subjects, dimensions, filters, and calculations. When you open a package in Cognos 8, these model objects are visible in the left frame.

The package must be previously created and published to the Cognos Connection portal. For more information, see the Framework Manager *User Guide*.

Steps

1. In the Cognos Connection Welcome page, click the **Launch** link.

   From the drop-down list, click **Report Studio**.

2. In the **Select a package** dialog box, click the package you want to use.

3. In the **Welcome** dialog box, choose whether to open a new or existing report or template:

   - To create a new report or template, click **Create a new report or template** and choose a report template.

     Tip: You can specify a language other than the default language for your package by clicking on the ellipsis (...) button to the right of the **Package** field, clicking the **Language** ellipsis button in the **Report Package** dialog box, highlighting the desired language, and then clicking OK. As stated on the language selection dialog, items such as separators and decimals may need to be manually updated for language-specific syntax rules as a result of your selection.

   - To open an existing report or template, click **Open an existing report or template** and select a report.

Objects from the selected package, such as data items, appear on the **Source** tab of the **Insertable Objects** pane.

Tip: You can later change packages (p. 243).

Refresh the Package

If the package that a report is using has changed, refresh it to ensure that you are working with the latest version.

Steps

1. In the **Insertable Objects** pane, click the **Source** tab .

2. Right-click its background and click **Refresh**.

Tip: You can also close and reopen the report to upgrade it to the latest version of the package.

Choose a Report Template

When creating a new report, choose a report template to get you started quickly. You can select a predefined template that is formatted as a particular report type (p. 79), or you can open an existing report as a template.

Steps

1. From the **File** menu, click **New** .

2. If you want to change the package, click the ellipsis (...) button and click a different package.
3. Choose a report template:
   - To create a report using one of Report Studio's predefined templates, double-click the template you want.

     Tip: Double-click **Report Template** to create a report template that can be applied to reports created in Query Studio or Analysis Studio.
   - To create a report using a previously created template, double-click **Existing**, locate the template, and click **Open**.

     Tip: Click the **Type** box and then click **Templates** to see only existing templates.
   - To open an existing report as a template, double-click **Existing**, locate the report, and click **Open**. You are actually opening a copy of the report, and the original report remains unchanged.

The report template appears in the report page. For more information about templates, see "Creating Your Own Report Templates" (p. 74).

Add Data to a Report

Select the data items that you want to appear in the report.

If you frequently use items from different query subjects or dimensions in the same reports, ask your modeler to organize these items into a folder or model query subject and then to republish the relevant package. For example, if you use the product code item in sales reports, the modeler can create a folder that contains the product code item and the sales items you need.

Data Source Icons

Each object in the data source has a representative icon. You can insert all of the following objects in a report, except for packages and dimensions.

Icon	Object
	Package, which contains the objects you can insert in a report.
	Query subject, which represents a table in the database.
	In relational data sources, query item, which represents a column of qualitative data in the database, such as product name or country. In dimensional data sources, level attribute, which represents a property of a level.
	A member is a unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products Hierarchy.

Icon	Object
	Dimension, which represents a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets.
	Hierarchy, which represents a collection of dimensional members organized into a tree structure.
	Level, which is a set of members that have common attributes. For example, a geographical dimension might contain levels for country, region, and city. Multiple levels can exist within a level hierarchy, beginning with the root level. The root level is the parent and rollup of all members in the first level. It is used to obtain a rollup of all values across the hierarchy and to provide a convenient point to start drilling. For example, a Years level hierarchy may contain the following levels: • Root level Years • First level Year • Second level Quarter • Third level Month
	Measure or fact, a query item that represents a column of quantitative data in the database, such as revenue or quantity.
	Measures folder, which contains hierarchical measures.
	Model filter.
	A folder is used to organize data items. You cannot import a folder into your report.

Step

- In the **Insertable Objects** pane, on the **Source** tab , drag each data item to the location where you want it to appear.

  A flashing black bar indicates where you can drop an item. Items inserted in the report appear on the **Data Items** tab .

Tips

- For more information about a data item, right-click the item and click **Properties**. For example, when working with relational data sources, the **Usage** property identifies the intended use for the data represented by the data item.

- Other ways to select data items are to double-click each item, or right-click each item and click **Insert**.
- If you want to remove a data item from the report, select it and click the delete button . To remove the data item from the report but keep it on the **Data Items** tab, click the cut button instead.

Customize Your Source Tree

When using dimensional data, you can customize how you see data in the source tree. You can select a metadata view with dimensions, hierarchies, levels, an optional members folder, and member properties.

Or you can select a members-oriented tree view, such as the one in Analysis Studio. This view allows you to add only members in your report.

Steps

1. In the **Insertable Objects** pane, click the **Source** tab .
2. Right-click anywhere in the pane and click **Package Tree Settings**.
3. Select the options that you want and click **OK**.

Save a Report

Save your report to preserve the modifications you made.

Reports are saved to the Cognos 8 server. You can also save your report on your computer (p. 235).

Steps

1. From the **File** menu, click **Save** , or click **Save As** to save a copy of the report under a different name.
2. If you are saving the report for the first time, specify where you want to save the report and type a file name.

   For information about setting up folders in Cognos Connection for your reports, see the Cognos Connection *User Guide*.
3. Click **Save**.

Run a Report

Run your report to see the data that is retrieved. Save time by validating it first to check for errors.

You can also run a report or a group of reports in Cognos Connection. Furthermore, you can save report outputs to a file system. For more information, see the *Administration and Security Guide*.

You can specify not to render a report page if it does not contain any data (p. 70).

If you run a report that uses functions or features not supported by the data source, an error message appears. We recommend that you periodically test your reports while you author them in Report Studio to ensure that you do not encounter multiple error messages when you run the report.

Steps

1. Open the report that you want.
2. If you want to clear parameters values stored on the Cognos 8 server, from the **File** menu, click **Clear Parameter Values**.

   Parameter values stored by the Cognos 8 server include signon, validation, and prompt information. For example, if you define two data source connections in Cognos Connection that point to the same data source, you will be prompted to choose one when you run a report. This information is stored so that you will not be prompted each time you run the report.
3. From the **Tools** menu, click **Validate Report** .

   A message box appears indicating whether any errors were found in the report.
4. If you require more detail from the validation process, revalidate the report by clicking **Validate with Options**, and do the following:
   - Click one of the following validation levels.

Validation level	Description
Error	Retrieves all errors returned from the query.
Warning	Retrieves all errors and warnings returned from the query.
Key Transformation	In addition to errors and warnings, retrieves informational messages describing important transformation steps from the report specification to the native query sent to the data source. These messages can be helpful to understand the cause of errors and warnings returned from the query.
Information	Retrieves errors, warnings, key transformations, and other information related to query planning and execution.

   - Select the **Interactive Data** check box to specify that no query optimization is to be used.

     The **Interactive Data** check box controls how queries will be processed during validation.

     Clear the **Interactive Data** check box to set the **Execution Optimization** property to **All Rows**.

     Tip: For more information about the **Execution Optimization** property, see Execution Optimization in "Report Studio Object and Property Reference" (p. 523).

     A message box appears indicating whether any errors were found in the report.

5. If you encounter validation errors and want Report Studio to identify incorrect objects in your report, from the **Tools** menu, click **Auto Correct**.

   Report Studio provides a list of such objects, each of which you can then remove to successfully run your report.

6. If you want to view only the tabular data, from the **Run** menu, click **View Tabular Data** .

   If the report contains multiple queries, you must first click an object, such as a list or crosstab, that uses the query for which you want to view the tabular data.

   Use this command to ensure that the right results appear. For example, you create a calculation (p. 189) and you want to ensure it is giving you the results you want.

   Tip: You can also view tabular data in Query Explorer (p. 213), which is useful when you are building queries.

7. If you want to set run options, from the **Run** menu, click **Run Options**.

   The default value is the value of the selected corresponding run option in Cognos Connection.

   The run options you set apply only to the current session. When you close Report Studio, the options return to the default settings.

8. Change any values you want for the current session and click **OK**.

Option	Notes
Format	None.
Paper size	Specify only if the output format is PDF.
Paper orientation	Specify only if the output format is PDF.
Data mode	Specify how much data is returned. **All Data** returns all data. **Limited Data** limits the amount of data returned based on design mode filters defined in the package. **No Data** returns artificial data instead of actual data from the data source. For more information about design mode filters, see the Framework Manager *User Guide*.
Language	The content language sets the preferred language for the data, Cognos Viewer, dates, and so on.

Option	Notes
Rows per page	Specifies the number of rows to appear on each page. A **Rows Per Page** property exists in the **Properties** pane for lists and crosstabs. If you specify a value for this property, it overrides the same-named run option. This property applies to both HTML and PDF outputs. For more information about this property, see "Controlling the Rows Per Page for Multiple Containers in HTML and PDF" (p. 70).
Prompt	Select to be prompted (p. 199) for each prompt defined, unless the prompt is defined in a report page. If you clear the check box, you are prompted only if the report cannot run without user intervention. For example, if a report has a single parameterized filter that is optional, you are not prompted when you run the report.

9. From the **Run** menu, click one of the options to produce the report in the format you want.

   You can produce a report in HTML, PDF, CSV, various Excel formats, and XML. You cannot produce a report in CSV or XML format if you have more than one query defined in the report, unless the additional queries are used for prompts.

The report runs in Cognos Viewer. Once the report has finished running, you can run the report again in the same format or in a different format. If you run the report again in CSV or XLS format, the report will appear in a new browser window.

The options available in Cognos Viewer depend on the capabilities set by the administrator for each user. For more information, see the *Administration and Security Guide*.

Running a Report Against a Dimensional Data Source

You can cancel a report that is running against Microsoft SQL Server Analysis Services only during the initial portion of its execution. After this time, the report runs to completion.

The same behavior applies to SAP BW data sources.

In Framework Manager, you can also control the number of levels within a hierarchy from which members, or values, are extracted from the hierarchy to populate a tree prompt.

For SAP BW, you can reduce the number of hierarchy levels to limit the number of nodes by setting the SAP BW variable property `trimHierarchyLevels` to 1. This removes the lowest level from the hierarchy prior to creating the list of nodes.

Units of Measure Notation

When running a report against an SAP BW data source, units of measure are included in the same column as the data values, separated by one space. For example, Celsius and Fahrenheit notations are appended to the end of the value.

If you see an asterisk character (*), one of the following was detected:

- an unknown currency
- a value of unknown or questionable unit of measure, such as a mixed currency calculation or rollup

  Mixed currency values occur when you calculate values with different currencies.

This behavior occurs when you are using a Cognos cube as a data source.

This behavior also occurs for SAP BW data sources.

Unsupported SAP Variable Properties

Some SAP variable properties are not supported.

Exclusionary ranges appear as an inclusionary prompt.

Mandatory not initial appear as a mandatory prompt.

When using BEx to define variables in your SAP data source, avoid using exclusionary ranges and the mandatory not initial property.

Specify Not to Render a Page If It Does not Contain Data

You can specify not to render a report page if the page does not contain any data when the report is run.

Steps

1. In the report page, click a data container.
2. In the **Properties** pane, click the select ancestor button and click the data container type.

   For example, if the data container is a list, click **List**.
3. Set the **Render Page when Empty** property to **No**.
4. Repeat steps 1 to 3 for all other data containers in the page as well as any table of contents objects (p. 143).

When you run the report, if no data is produced in all data containers and table of contents objects in a page, the page is not rendered. The page is not rendered even if the page contains other objects, such as text items or images.

Controlling the Rows Per Page for Multiple Containers in HTML and PDF

If you have more than one data container in a report, such as a list and a crosstab, you can control how the report is rendered in HTML and PDF by setting the **Rows Per Page** property for each container.

For HTML output, the report property option **Page break by data container for interactive HTML** (p. 164) controls whether the default number of rows is rendered for each data container on each page.

Cognos 8 uses the following rules when rendering reports in HTML and PDF:

- If the **Rows Per Page** property is not set for any of the data containers, 20 rows per page are rendered in HTML and each page is completely filled in PDF. The first data container is rendered until there is no more data, followed by the next container, and so on.

  Tip: The number of rows that appear on a PDF page depends on the font size set in the report.

- If the **Rows Per Page** property is set for each data container, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data.

- If the property is set for only some of the containers, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data. For the remaining containers, 20 rows per page are rendered on each page in HTML and each page is completely filled in PDF.

  For example, you have two lists, List1 and List2. You set the **Rows Per Page** property to 5 for List1. When you run the report in HTML, the first page contains the first 5 rows from List1 followed by the first 15 rows of List2.

- If no data is returned for a data container, an empty container is rendered.

Producing a Report in CSV Format

Cognos 8 can produce reports in CSV format so you can open them in other applications, such as Microsoft Excel. Reports saved in CSV format

- are designed to support Unicode data across many client operating systems
- are UTF-16 Little Endian data encoded
- include a BOM (Byte Order Mark) at the beginning of the file
- are tab-delimited
- do not enclose strings in quotation marks
- use a new line character to delimit rows

You can open reports saved in CSV format using Microsoft Excel, Microsoft Windows Wordpad, and Star Office. By default, reports produced in CSV format will appear in the application associated with the .csv file type.

You cannot produce the following in CSV format:

- maps
- charts that do not have at least one category or series
- reports that have more than one query defined in the report, unless the additional queries are used for prompts.

In Cognos Connection, you can configure the CSV output to suit your environment. For example, you can specify the character used to delimit fields. For more information, see the *Administration and Security Guide*.

Producing a Report in Excel Format

Cognos 8 can produce reports in Excel format. Three options are available:

- **Excel 2000 Single Sheet** will produce reports on one sheet that you can view in Microsoft Excel versions earlier than 2002.
- **Excel 2002** will produce reports that you can view in Microsoft Excel versions earlier than 2007.
- **Excel 2007** will produce reports that you can view in Microsoft Excel version 2007.

The Excel 2000 format can be viewed with versions of Excel 2000 and later. It supports up to 65,536 rows and multiple sheets. The Excel 2000 format is only available in Cognos 8 when Cognos Application Firewall validation is disabled.

Excel 2000 single sheet format offers improved security. Excel 2000 may have cookies in the URLs to spreadsheets, which could be used to illegally impersonate a user. Excel 2000 single sheet format does not use outgoing URLs. However, there is a limit of 65,536 rows, and page breaks for multiple-author pages are ignored.

Excel 2002 format and Excel 2000 single sheet format also offer the following benefits:

- Both work with SSL protocol.
- Both work with a single signon.

  Secure reports can be accessed without subsequent signons because the system automatically identifies users and provides security information.
- Both work with Netscape 7.01.
- Spreadsheets are contained in a single file for reliable spreadsheet navigation.

The Cognos 8 full date format does not always appear correctly in Microsoft Excel 2000 if it is on a system with a regional setting other than English. This does not occur with Microsoft Excel 2002 or 2003 versions. To fix this, reapply the proper format in Excel.

Excel 2007 format renders report output in native Excel XML format, also known as XLSX, that provides a fast way to deliver native Excel spreadsheets to Microsoft Excel XP, Microsoft Excel 2003, and Microsoft Excel 2007. Users of Microsoft Excel XP and Microsoft Excel 2003 must install the Microsoft Office Compatibility Pack, which provides file open and save capabilities for the new format. The output is similar to other Excel formats, with the following exceptions:

- Charts are rendered as static images.
- Row height can change in the rendered report to achieve greater fidelity.

Limitations exist when producing reports in XLS format. For more information, see "Producing Reports in Microsoft Excel Format" (p. 515).

Producing a Report in XML Format

XML report outputs save the report data in a format that conforms to an internal schema, xmldata.xsd. You can find this schema file in *c8_location*/bin.

This format consists of a dataset element, which contains a metadata element and a data element. The metadata element contains the data item information in item elements. The data element contains all the row and value elements.

You can create models from reports and other data that conform to the xmldata.xsd schema. This is useful if you want to use a report as a data source for another report, or if you use a database that cannot be read by Framework Manager. In this case, export the data from the data source to an XML file, in conformance with the xmldata schema, and then open the XML file in Framework Manager.

For more information, see the Framework Manager *User Guide*.

You cannot produce the following in XML format:

- maps
- charts that do not have at least one category or series
- reports that have more than one query defined in the report, unless the additional queries are used for prompts.

Set PDF Page Options

Set PDF page options to control how report pages appear in PDF. You can set PDF page options for individual report pages or for all report pages in a layout (p. 264).

You can also set PDF page options in Cognos Connection. For more information, see the Cognos Connection *User Guide*.

Steps

1. To set PDF page options for all report pages in a layout, do the following:,
   - From the **File** menu, click **PDF Page Setup** and set the page options that you want.
   - Click **OK**.
2. To set PDF page options for an individual page, do the following:
   - Pause the pointer over the page explorer button and click the page that you want.
   - In the work area, click anywhere on the page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.
   - Double-click the **PDF Page Setup** property and set the page options that you want.
   - Click **OK**.

Creating Your Own Report Templates

A report template is a pattern you use to build reports. Create your own report templates when you frequently produce the same type of report. A template can include the following objects:

- HTML items
- hyperlinks
- page numbers
- images
- tables
- text items
- blocks
- layout calculations
- date
- time

For more information about these objects, see "Formatting a Report" (p. 133).

To create a report template, you can

- convert a report to a template
- create a new template

For more information about how to create a report template, see the Report Studio Tour. In addition, you can use sample templates provided with Report Studio.

Tip: You can add your own report templates to the **New** dialog box. For more information, see the *Administration and Security Guide*.

Convert a Report to a Template

Convert a new or existing report to a template so it can be reused.

Steps

1. Create a new report or open an existing report.
2. Add the objects you want to the work area.
3. From the **File** menu, click **Convert To Template**.

   Any query-related information in the original report, such as data items, calculations, and filters, is removed from the template.
4. From the **File** menu, click **Save As** to save the template as a new file and keep the original report intact.

Create a New Template

Create a new template to provide Query Studio and Analysis Studio users with a layout that they can use to create reports or apply to existing reports. You can also use the template in Query Studio to define a layout for prompt pages.

Steps

1. From the **File** menu, click **New**.
2. Click **Query Studio Template** and click **OK**.
3. Pause the pointer over the page explorer button and click the report page or prompt page you want to format.

   Tip: To create a new report page or prompt page, click the **Report Pages** or **Prompt Pages** folder.
4. In the **Insertable Objects** pane, click the **Toolbox** tab.
5. Add the objects you want to the work area.

   If you add objects that are not supported by Query Studio, the objects will be ignored when you apply the template.
6. Save the template.

Techniques for Creating Accessible Reports

Creating accessible reports ensures access of information to all users, with all levels of ability.

For example, blind users may use screen reading technology to access the information in a report.

The following are some design considerations for creating accessible reports:

- Avoid using visual cues, such as text bolding or color, to convey important information.
- Avoid using pictures and OLE Objects in PDF documents, as these items are tagged as artifacts and ignored by the screen reader.
- Avoid using conditional formatting to convey important information.
- Because chart types are rendered as images, which the screen reader ignores, ensure that there is a corresponding table with the same information.
- Deliver reports through HTML, as it is the most supported output format for most screen readers.
- Always ensure there is a title on the report.
- Gain an understanding for screen reading technology.
- Avoid spelling and grammar errors, as they cause the screen reading software to misinterpret the information.

- Although using prompts is fine, such as check boxes, radio buttons, combo boxes, and multi-select boxes, avoid using features like calendar boxes and up and down selections on time controls.
- When choosing to use embedded Web applications or drill-through paths, ensure the target application is also accessible.
- Avoid using large, complex list or crosstab reports.

  Displaying the information in multiple simple lists or crosstab reports is more manageable for users of Assistive Technology.

Techniques for Creating Reports for Cognos 8 Go! Office

Cognos 8 Go! Office provides an integrated environment for Cognos products and Microsoft Office. You can use Cognos 8 Go! Office to select pieces of reports to embed in Microsoft Excel workbooks, Microsoft Word documents, or Microsoft PowerPoint presentations, including data, metadata, headers, footers, and charts. You can use predefined reports or you can create new content using PowerPlay Web, Query Studio, or Report Studio.

Note that to access PowerPlay content, your administrator must configure PowerPlay to work with Cognos 8. PowerPlay content that is published only to Upfront is not available to Cognos 8 Go! Office.

Because Cognos 8 Go! Office cannot fully convert highly formatted reports into Excel or other Microsoft document output types, you may not get the results that you want.

To create effective reports for Cognos 8 Go! Office, follow these recommendations:

- Create content to meet specific Microsoft Office integration needs.

  For example, in Cognos 8, many options are available to format data. Use less formatting to make data more accessible to Office applications.
- Organize reports.

  You can publish workbooks to Cognos Connection and organize them with your reports in **Public Folders** or **My Folders**. For more information, see the Cognos Connection *User Guide*. By organizing your content, you can quickly retrieve the information that you want.

  Tip: Workbooks, documents, and presentations that are enabled for Cognos 8 Go! Office are identified by their own unique icons, helping you to distinguish them from other types of files.
- Optimize report templates for Microsoft Office.

  If you rely on IT personnel or other report authors to create content, request report templates that are optimized for your Microsoft Office integration needs. You may want to request only the data elements or queries that you need and request minimal formatting so that you can more easily use Microsoft Office formatting capabilities with the Cognos content. For example, reports authored in Report Studio can contain list objects embedded within list objects with specific formatting applied. When converted to the tabular representation available in Excel, these reports may not be rendered in the same way in which they appear in Cognos 8.

- Format elements in the Office application.

  Instead of formatting objects in Cognos 8, add the formatting that you want in the Office application. By applying less formatting in Cognos 8, more data can be imported into the desired locations.

- Label report elements using descriptive names.

  This practice makes them more easier to find after you import them. Examples of report elements include lists, crosstabs, and charts.

- Do not nest report objects.

  If you nested report objects, some objects may not appear in the correct location, or they may not appear at all. In addition, nesting report objects may cause the following error message to appear:

  RDS-ERR-1000 Report Data Service could not process from the content provider.

  For example, this error occurs if a repeater or repeater table is inside a block or table. It also appears layout objects, such as lists, crosstabs, and charts, are in a conditional block that is inside another block or table.

- Keep table sizes small.

  For example, because of the size of slides, the maximum number of rows and columns that you can have in PowerPoint tables is 25. Although Word and Excel permit larger tables, it takes more time to download and render them.

- Use images with transparent backgrounds.

  The background will show through the image in the Office application, making the image look like part of the presentation. If you want, you can then supply your own background color.

- If you have an image in a list, specify its height and width.

  This practice ensures that the image appears in the correct size in the Office application.

- Remember that graphs and charts are imported as images.

  Images in Cognos 8 have image maps associated with them to enable tooltips and hotspots. Cognos 8 Go! Office cannot import tooltips and hotspots into Office applications.

- Consider the additional limitations that exist when producing reports in Excel format (p. 515).

Managing Your Reports

After you create one or more reports, you can manage them for yourself and others. You can perform the following tasks in Cognos Connection:

- Schedule a report or a group of reports to run at a specific time.
- Distribute reports to other users.
- Print a report.

- Select the language used when a report is run.
- Set prompt values.
- Maintain the history of a report.
- Maintain different versions of a report.
- Create report views.

For information about how to perform these tasks, see the Cognos Connection *User Guide*.

The Cognos 8 SDK

When you create a report in Report Studio, you are creating a report specification. A report specification is an XML file that you can view (**Tools** menu, **Show Specification**). In addition, you can view the specification for a selected object with the **Show Specification** (**Selection**) menu option.

Tip: When you are viewing the report specification in Report Studio, you cannot modify or copy parts of it.

Instead of using Report Studio, you can programmatically create or modify reports by using an editing tool to work with report specifications. You then use the Cognos 8 software development kit (SDK) to implement the reports in your Cognos 8 environment. This is useful if, for example, you must make the same modification in many reports. Rather than opening each report in Report Studio and making the change, you can automate the process using the SDK, thereby saving you time. For more information about the Cognos 8 SDK, contact your local Cognos sales office.

Tip: You can also modify the XML code in a report specification by saving the report specification on your computer (p. 235).

Chapter 3: Types of Reports

You use different report types to look at data in different ways or to answer different business questions. For example, you create a list report to show your entire customer base, but you create a crosstab report to show which sales representative has the highest sales for each product line. You can create a chart to present data graphically, and use a repeater to create mailing labels.

In Report Studio, you can create the following types of reports:

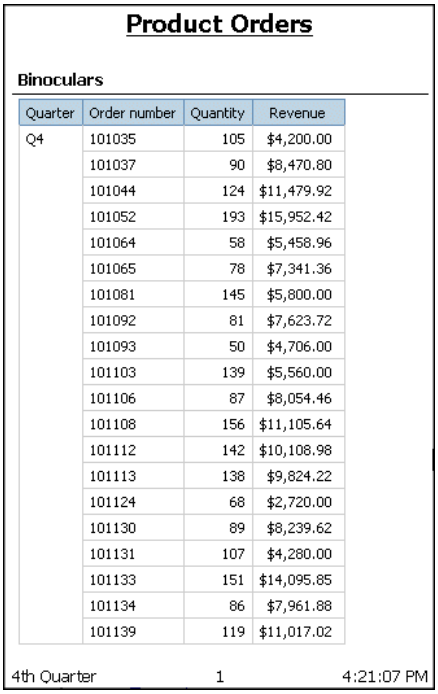
- list
- crosstab
- chart
- map
- repeater

You can also combine one or more of these in a single report.

List Reports

Use list reports to show detailed information from your database, such as product lists and customer lists.

A list report is a report that shows data in rows and columns. Each column shows all the values for a data item in the database or a calculation based on data items in the database.



Product Orders

Binoculars

Quarter	Order number	Quantity	Revenue
Q4	101035	105	$4,200.00
	101037	90	$8,470.80
	101044	124	$11,479.92
	101052	193	$15,952.42
	101064	58	$5,458.96
	101065	78	$7,341.36
	101081	145	$5,800.00
	101092	81	$7,623.72
	101093	50	$4,706.00
	101103	139	$5,560.00
	101106	87	$8,054.46
	101108	156	$11,105.64
	101112	142	$10,108.98
	101113	138	$9,824.22
	101124	68	$2,720.00
	101130	89	$8,239.62
	101131	107	$4,280.00
	101133	151	$14,095.85
	101134	86	$7,961.88
	101139	119	$11,017.02

4th Quarter 1 4:21:07 PM

For information about how to create a list report, see the Report Studio Tour.

Group Data

Group data items in a list report to remove duplicate values. For example, you have a report that shows products purchased. For each product, the product type is also shown. You group the Product type column to show only one instance of each product type in the list.

You can also group data items in repeaters (p. 130) and in page sets (p. 152).

In addition to grouping, you can specify the sort order of data items. In lists, you can sort data items within groups as well as ungrouped items. In crosstabs, you can sort row and column items. For more information, see "Perform Advanced Sorting" (p. 179).

If you are querying a dimensional data source, you can group or sort in the order of the existing hierarchy only. If you change the order, an error such as the following occurs.

OP-ERR-0199: The query is not supported. The dimensions on the edge are inconsistent. The dataItems from dimension="[Locations]" must be adjacent.

You can omit columns from the grouping. For example, if a level hierarchy contains the levels Country, State, and City, and another level hierarchy contains the level Product:

- Country, State, City is valid.
- Country, Product, and State is not valid because a level from another level hierarchy was inserted between two levels from another level hierarchy.
- Country, City, and Product is valid even though State is omitted.
- Country, Country code is valid, where Country code is a member property of Country.

To override the hierarchical order that the data source specifies, you can override the dimension information of the query. For more information, see "Add Dimension Information to a Query" (p. 223).

We recommend that you do not use member properties of a dimension to group data. Grouping by using member properties may give unexpected results, such as duplicates.

Steps

1. Click the column you want to group on.

   You can click either the column heading or one of the column cells.

   Tip: To perform multiple groupings at once, use Ctrl+click or Shift+click.

2. From the **Structure** menu, click **Group/Ungroup**.

A symbol appears indicating that the column is grouped. You can control when its values appear by setting the group span, and the column appears in the list of headers and footers that you can add to the report (p. 135).

Note: Grouping all columns in a list is equivalent to creating a single-edge crosstab (p. 87).

Perform Advanced Grouping

You can perform more advanced groupings in a list to get the results you want. For example, you can

- view the entire grouping structure and make changes to the grouping order
- group a data item that appears in a query but not in the layout
- specify a list of properties (p. 197) for a group

You can also perform advanced sorting (p. 179).

Steps

1. Click a column in the list.
2. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.
3. In the **Properties** pane, double-click the **Grouping & Sorting** property.
4. If you want to change the grouping order, in the **Groups** pane, click the grouped item that you want in the **Groups** folder and drag to the location that you want in the grouping hierarchy.
5. If you want to create a new grouping, in the **Data Items** pane, click the data item that you want and drag it to the **Groups** folder in the **Groups** pane.
6. If you want to specify a list of properties for a group, do the following:
   - Click the group that you want.

     Tip: Click **Overall** to specify a list of properties for the overall group. The overall group is the grouping of all data into one group. You can specify a list of properties for the overall group to add overall totals to the list.

   - Click the properties button .
   - Select the appropriate check boxes.
   - Click **OK**.
7. Click **OK**.

Set the Group Span for a Column

When columns are grouped, you can choose how often to show column names by changing the group spanning. For example, when country and city are both grouped, you can choose to show the country name each time

- the country changes, by spanning Country by Country
- the city changes, by spanning Country by City
- there is a new record, by specifying no spanning

Spanning one grouped column by another column is helpful if the second column contains many items.

Steps

1. Click the column for which you want to set the group span.
2. In the **Properties** pane, click the **Group Span** property and click the column you want to span.

Example - Remove Identical Values in a List

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all product orders, organized by order number. To make the report easy to read, you group the Order number column so that only one instance of each order number appears. However, because each order contains different products, the same order date appears for each product. You decide to show the order date only when the order number changes.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse** (**query**) link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing** (**query**) and **Sales** (**query**).
6. Expand **Sales order** and add **Order number** by double-clicking it.
7. Expand **Time dimension** and add **Date** by double-clicking it.
8. Expand **Product** and add **Product name** by double-clicking it.
9. Expand **Sales fact**. Add **Quantity** and **Revenue** by double-clicking them.
10. Group the **Order number** column.
11. Click the **Order date** column.
12. In the **Properties** pane, click the **Group Span** property and then click **Order number**.

When you run the report, the first row of the **Order date** column appears for each **Order number** row.

Order number	Date	Product name	Quantity	Revenue
100001	Jan 12, 2004	Flicker Lantern	256	8,624.64
		Polar Ice	92	9,411.6
100002	Jan 12, 2004	Bear Edge	172	6,690.8
		Edge Extreme	162	18,032.22
		Glacier GPS Extreme	74	24,747.82
		Insect Bite Relief	422	2,532
		Mountain Man Deluxe	90	6,825.6
100003	Jan 12, 2004	BugShield Extreme	3,252	21,170.52
		Polar Extreme	19	2,733.15
		Seeker 50	88	10,975.36
		Sun Shield	1,107	6,376.32
100004	Jan 12, 2004	Hibernator Lite	354	29,658.12
		Star Gazer 2	139	75,289.35
		Star Lite	261	89,841.42
		TrailChef Deluxe Cook Set	279	33,658.56
100005	Jan 12, 2004	Bear Survival Edge	120	10,299.6
		Double Edge	382	6,165.48
		Lady Hailstorm Titanium Woods Set	26	32,878.04
		Mountain Man Combination	44	4,139.96
		Mountain Man Extreme	24	6,711.84

Example - Repeat a Column Value in a List

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all products sold by the company, organized by the product line and product type. To make the report easier to read, you group the Product line and Product type columns so that only one instance of each column appears. However, because some product lines contain many different product types and products, you decide to show the product line for each product type.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)**.
6. Expand **Sales (query)** and **Product**. Add the following data items by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**
7. Expand **Sales fact** and add **Product cost** by double-clicking it.
8. Group the **Product line** and **Product type** columns.
9. Click the **Product line** column.
10. In the **Properties** pane, click the **Group Span** property and click **Product type**.

11. Click any part of the list and then click the select ancestor button in the title bar of the **Properties** pane.
12. Click **List**.
13. In the **Properties** pane, double-click the **Font** property.
14. In the **Size** box, click **8pt** and click **OK**.

When you run the report, the product line appears whenever the product type changes.

Product line	Product type	Product name	Product cost
Personal Accessories	Binoculars	Seeker Mini	6,914,040.00
		Seeker 50	14,785,118.58
		Seeker Extreme	10,560,169.88
		Ranger Vision	21,148,795.52
		Seeker 35	21,104,631.45
		Opera Vision	4,132,887.98
Mountaineering Equipment	Climbing Accessories	Granite Chalk Bag	1,723,827.70
		Firefly Charger	6,755,269.04
		Granite Belay	8,961,338.25
		Granite Pulley	7,227,000.70
		Granite Carabiner	6,166,540.24
		Firefly Climbing Lamp	4,562,980.34
		Firefly Rechargeable Battery	4,197,897.90
Camping Equipment	Cooking Gear	TrailChef Cup	1,545,089.31
		TrailChef Water Bag	11,292,005.38
		TrailChef Kettle	11,373,514.16
		TrailChef Utensils	8,611,463.50
		TrailChef Kitchen Kit	13,608,439.00
		TrailChef Deluxe Cook Set	35,527,805.84

Format Lists

Format lists to give them the appearance you want. You can specify formatting for individual column titles or bodies or for all columns in a list. When you specify formatting for all columns in a list, the formatting is automatically applied to new columns you subsequently add.

You can also quickly format lists by applying table styles (p. 147).

Steps

1. To format an individual column, do the following:
   - Click the column title or column body.

     To format the text in the column title or the data in the column body, click the unlock button in the toolbar and then click the title or body.

     To format the whole column, click the select ancestor button in the title bar of the **Properties** pane and click **List Column**.
   - In the **Properties** pane, click the property you want and specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
2. To format all columns, do the following:
   - Click a column in the list.

- To format list column bodies, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Body Style**.
- To format list column titles, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Title Style**.
- To format entire columns, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns**.
- To format the entire list, click the select ancestor button in the title bar of the **Properties** pane and click **List**.
- In the **Properties** pane, click the property you want and specify a value.

For example, to specify a background color, click **Background Color** and choose the color you want to use.

Formatting for specific columns overrides formatting for entire columns. For example, you specify red as the background color for a specific column body and green as the background color for all columns. When you run the report, the specific column body is red and the remaining columns in the list are green. New columns added to the list will have green as their background color.

For more information about formatting reports, see "Formatting a Report" (p. 133).

Crosstab Reports

Use crosstab reports to show information in a more compact form than in a grouped list. For example, create a crosstab report to show total sales by product line generated by each sales representative.

Like list reports, crosstab reports are reports that show data in rows and columns. However, the values at the intersection points of rows and columns show summarized information rather than detailed information.

Profitability by Product Line

		2007			2006	
		Gross profit	Revenue	*Cost of Goods Sold*	Gross profit	Reven
Personal Accessories	Glacier Deluxe	1,741,826.22	3,928,757.22	*2,186,931*	1,669,390.15	3,958
	Glacier GPS Extreme	5,639,265.95	12,034,715.22	*6,395,449.27*	5,921,887.95	12,249
	TX	8,306,303.78	18,172,476.5	*9,866,172.72*	14,701,866.32	32,35
	Pocket Gizmo	2,675,373.35	4,380,582	*1,705,208.65*	3,454,153.34	5,68
	Seeker 35	2,235,849.84	7,754,925.78	*5,519,075.94*	2,574,835.2	8,93
	Zone	9,434,315.06	28,021,707.95	*18,587,392.89*	16,263,229.54	49,62
	Mountain Man Deluxe	2,269,799.48	4,563,194.48	*2,293,395*	2,493,495.99	4,972
	Polar Extreme	347,639.66	701,512.16	*353,872.5*	499,545.53	998
	Ranger Vision	5,622,674.4	10,525,086.6	*4,902,412.2*	5,891,875.71	11,71

For information about how to create a crosstab report, see the Report Studio Tour.

Crosstab Nodes and Crosstab Node Members

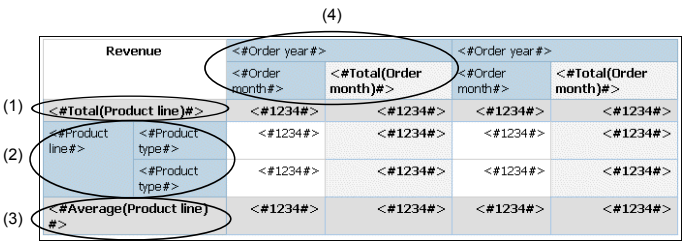
When you add data items to crosstabs, you create crosstab nodes and crosstab node members. These objects allow you to easily create crosstabs, using drag-and-drop operations.

Crosstabs are dimensional objects that have row edges and column edges. Each edge is composed of a set of crosstab nodes. Each crosstab node contains the following:

- One or more crosstab node members.
- Zero or one nested crosstab node, which contains one or more crosstab node members or nested crosstab nodes.

Each crosstab node member refers to a data item that contains an expression that defines the members that appear in the crosstab.

The following crosstab contains four crosstab nodes.



Crosstab node 1 contains a single node member for the total. This node refers to the data item **Total(Product line)**.

Crosstab node 2 contains a crosstab node member that refers to the data item **Product line**. This member has a nested crosstab node containing a crosstab node member that refers to the data item **Product type**.

Crosstab node 3 contains a single node member for the average. This node refers to the data item **Average(Product line)**.

Crosstab node 4 contains a crosstab node member that refers to the data item **Order year**. This member has two nested crosstab nodes. The first node contains a crosstab node member that refers to the data item **Order month**. The second node contains a crosstab node member for the total. This node refers to the data item **Total(Order month)**.

Tip: Nodes are also created when you add data items to charts.

Moving Crosstab Nodes

Crosstab nodes can be placed anywhere in the crosstab. For example, in the previous diagram, you can drag **Order month** under **Average(Product line)** to create a row edge.

To move nested items from one edge to another, ensure that you select the crosstab node and not the crosstab node member. For example, in the previous diagram, you want to move **Product line** and **Product type** to the column edge. To do this, click **Product line**, and in the **Properties** pane,

click the select ancestor button, and click **Crosstab Node**. Both **Product line** and **Product type** are selected.

Crosstab Node Creation Option

In the **Structure** menu, the **Create Crosstab Nodes** option affects the drag-and-drop behavior in crosstabs. When the option is turned on, and you add a data item to a crosstab, the item is created as a crosstab node. If the option is turned off, the item is created as a crosstab node member to an existing crosstab node.

For example, in the previous diagram, if the option is turned on and you drag **Country** from the **Insertable Objects** pane beneath **Product line**, **Country** becomes a new peer node to **Product line**. If the option is turned off, **Country** becomes a new peer node to **Product line** and has **Product type** as a nested crosstab node. This happens because instead of adding a new node, you are adding a new member to the existing node that already contains **Product line**.

Tip: To create discontinuous crosstabs (p. 93), turn the crosstab node creation option on. If you want the items on the edges of crosstabs to be related (contain the same nested items), turn the crosstab node creation option off.

Create a Single-Edge Crosstab Report

Create a single-edge crosstab report to show data in a list-like form. For example, you want a report that shows the quantity of products sold for each year and for each order method. Create a crosstab with **Order Year** and **Order Method** as rows and **Quantity** as the measure.

Steps

1. From the **File** menu, click **New**.
2. Click **Crosstab** and click **OK**.
3. In the **Insertable Objects** pane, on the **Source** tab, click the data item you want to add to the crosstab and drag it to **Rows** or **Columns**.

   A black bar indicates where you can drop the data item.
4. Repeat step 3 to insert additional data items.

   If you dragged the data item in step 3 to **Rows**, drag the additional items above or below the first item.

   If you dragged the data item in step 3 to **Columns**, drag the additional items to the left or right of the first item.
5. To add measures to the crosstab, drag the measures you want to **Measures**.

When you run the report, a crosstab is produced that has only one edge.

Create a Nested Crosstab Report

Nest data in a crosstab report to compare information by using more than one data item in a column or row. For example, a report shows the number of sales by product line for the past fiscal year. You decide to add a data item to further break down the number of sales by quarter.

When nesting columns in a crosstab report, there are four distinct drop zones where you can insert a new data item. The drop zone you choose will define the relationship between the data item and the column.

Rows

The following relationships are created when you insert a data item as a row:

- Inserting a data item to the left or right of a column creates a parent-child relationship between them.

  When you insert a data item to the left of a column, the data item becomes a parent to the column. When you insert a data item to the right of a column, the data item becomes a child of the column.

- Inserting a data item above or below a column creates a union relationship between them.

Columns

The following relationships are created when you insert a data item as a column:

- Inserting a data item to the left or to the right of a column creates a union relationship between the data item and the column.

- Inserting a data item above or below a column creates a parent-child relationship between them.

When you insert a data item above a column, the data item becomes a parent to the column. When you insert a data item below a column, the data item becomes a child of the column.

For example, you have a crosstab with Product line as rows and Quantity and Revenue as nested rows. For columns, you have Order method with Country as a nested column.

- Product line is a parent to Quantity and Revenue.

- Quantity and Revenue are peers.

- Order method is a parent to Country.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , click the data item you want to add to the report.

2. Drag the data item to the location in which you want it to appear as a nested column or nested row.

   A black bar indicates where you can drop the data item.

3. Repeat steps 2 to 3 to add other nested columns or rows.

   Tip: If you add more than one measure to a crosstab, all measures appear as columns. You cannot have one measure appear as a row and another as a column. To make all measures appear as rows, swap columns and rows (p. 157).

Format Crosstabs

Format crosstabs to give them the appearance you want. You can specify formatting for rows, columns, and fact cells in a crosstab, or for the entire crosstab. When you specify formatting for all rows, columns, fact cells, or the crosstab, the formatting is automatically applied to any new items you add.

Order of Styles Applied

If you apply the same styles, such as font color, to crosstab rows, columns, and crosstab intersections, the styles are applied in the following order:

- crosstab fact cells
- fact cells in the outermost rows
- fact cells in the innermost rows
- fact cells in the outermost columns
- fact cells in the innermost columns
- crosstab intersections

The style applied to the last object in the list overrides styles applied to previous objects. In addition, class styles are applied before styles that you apply manually.

You can also quickly format crosstabs by applying table styles (p. 147), and add white space to a crosstab by inserting crosstab space objects (p. 143).

Steps

1. To format the entire crosstab, do the following:
   - Click anywhere in the crosstab.
   - Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab**.
   - In the **Properties** pane, click the property you want and then specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
2. To format all rows, columns, or fact cells, click a row, column, or fact cell in the crosstab and then do the following:
   - If you clicked a row, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Rows**.

- If you clicked a column, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Columns**.
- If you clicked a fact cell, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Fact Cells**.
- In the **Properties** pane, click the property you want and then specify a value.

  For example, to specify a background color, click **Background Color** and choose the color you want to use.

  Tip: You can also right-click the row or column and click **Select Fact Cells**.

3. To format all crosstab cells for a specific row or column, do the following:
   - Click the row or column.
   - Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Member Fact Cells**.

     Tip: You can also right-click the row or column and click **Select Member Fact Cells**.
   - In the **Properties** pane, click the property you want and then specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
4. To format all row or column titles, do the following:
   - Click a row or column title.
   - Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Rows** or **Crosstab Columns**.
   - In the **Properties** pane, click the property you want and then specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
5. To format an individual row, column, or intersection, do the following:
   - Click the row, column, or intersection.

     Tip: To format the data in a row, column, or intersection, click the unlock button in the toolbar and then click the text item you want to format.
   - In the **Properties** pane, click the property you want and then specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.

Tip: In cases where fact cell formatting applied to rows conflicts with fact cell formatting applied to columns, you can use the **Fact Cells Precedence** property in the **Properties** pane to determine whether the row formatting or the column formatting has precedence. To specify this property, click anywhere in the crosstab. Then click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab**.

For more information about formatting reports, see "Formatting a Report" (p. 133).

Indent Data

You can indent crosstab node members to set them apart from surrounding data.

When you insert a hierarchy in the rows, all members are automatically indented according to their level. By default, the first member in a set is not indented. If you insert a hierarchy in the columns, the members are not automatically indented. You can change the indentation properties of members in the report.

You can also indent objects by applying padding (p. 148).

Steps

1. Click the crosstab node member that you want to indent.
2. In the **Properties** pane, select **Level Indentation**, and then click the ellipsis (...) button.
3. In the **Level Indentation** dialog box, specify your indentation type, length, and direction, and then click **OK**.

   Relative indentation means that the member will shift by one tab when the member's level increases relative to the previous member in the hierarchy.

   Indenting based on the level in the hierarchy means that the member will shift the number of tabs equivalent to the level in the hierarchy.

Change a List into a Crosstab

Change a list report into a crosstab report to view your data from a different perspective.

Steps

1. Click the columns that you want to appear as columns or nested columns in the crosstab.
2. From the **Structure** menu, click **Pivot List to Crosstab**.

The list becomes a crosstab, with the columns you selected in step 2 appearing as columns and nested columns. The unselected columns, except for measures, appear as rows and nested rows. If you have one measure, it becomes the cells of the crosstab. If you have more than one measure, they appear as columns.

Tip: To make all measures appear as rows, swap columns and rows (p. 157).

Example - Add Aggregate Data to a Crosstab Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales by order method to determine which methods are generating the most revenue and the highest sales volume.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse** (**query**) link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , add the following data items to the crosstab:
   - Expand **Sales and Marketing** (**query**) and **Sales** (**query**).
   - Expand **Product** and double-click **Product Line** to add it as rows.
   - Expand **Order Method** and double-click **Order Method** to add it as columns.
   - Click **Product type** and drag it just to the right of **Product line**.

     Product type is now nested in **Product line**.
   - Expand **Sales fact**. Drag **Quantity** to the **Measures area** to add it as a measure.
   - Drag **Revenue** to the right of **Quantity** to add it as a second measure.
6. Click any part of the crosstab and then click the select ancestor button in the title bar of the **Properties** pane.
7. Click **Crosstab**.
8. In the **Properties** pane, double-click the **Font** property.
9. In the **Size** box, click **8pt** and click **OK**.
10. Click one of the measures.
11. Click the aggregate button and click **Maximum**.

When you run the report, you can see that for camping equipment, **Cooking Gear** generated the highest sales volume for the **Special** order method, and **Tents** generated the most revenue. Tents ordered by the **Web** order method generated the largest revenue.

		E-mail		Fax		Mail		Sales visit		Special	
		Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue
Camping Equipment	Cooking Gear	667,254	12,660,160.31	222,711	4,375,855.61	171,358	3,349,528.79	1,435,032	29,955,819.16	115,031	2,344,882
	Lanterns	263,370	6,786,981.1	61,991	1,724,287.33	56,107	1,486,107.02	535,633	14,243,626.26	30,150	835,346.38
	Packs	151,719	18,578,756.65	38,840	5,737,253.68	34,452	4,260,591.65	276,738	35,504,440.57	14,940	2,659,194.73
	Sleeping Bags	174,310	14,711,086.99	48,580	5,894,604.37	39,916	3,997,819.92	325,265	35,363,204.56	20,638	2,806,666.82
	Tents	156,431	23,162,109.58	41,836	5,322,397.49	46,225	8,254,596.71	327,086	53,544,871.32	22,769	3,742,899.51
	Maximum (Product type)	**667,254**	**23,162,109.58**	**222,711**	**5,894,604.37**	**171,358**	**8,254,596.71**	**1,435,032**	**53,544,871.32**	**115,031**	**3,742,899.51**
Golf Equipment	Golf Accessories	194,588	3,446,977.94	60,831	1,064,185.81	44,488	835,353.31	157,959	2,594,946.04	27,454	398,608.96
	Irons	23,707	15,888,260.28	8,281	5,247,384.18	6,986	4,592,152.49	21,167	13,820,312.42	2,873	1,843,743.54
	Putters	95,756	8,490,684.25	26,600	2,213,771.51	23,720	1,921,430.83	67,412	5,645,827.58	6,309	629,444.44
	Woods	19,249	20,108,010.69	6,939	6,715,961.77	5,238	5,344,350.85	17,250	17,179,832.69	1,949	2,092,966.03
	Maximum (Product type)	**194,588**	**20,108,010.69**	**60,831**	**6,715,961.77**	**44,488**	**5,344,350.85**	**157,959**	**17,179,832.69**	**27,454**	**2,092,966.03**
Mountaineering Equipment	Climbing Accessories	138,242	2,038,855.77	172,803	2,390,328.78	46,488	679,297.57	598,362	8,432,511.02	63,772	802,217.59
	Rope	8,118	2,237,177.54	10,804	3,006,587.53	4,209	1,133,910	45,383	12,623,472.19	5,350	1,395,807.43
	Safety	17,881	1,283,131.17	36,195	2,658,809.76	8,065	592,500.66	125,179	9,265,767.61	8,386	625,119.3
	Tools	34,973	1,917,287.48	72,606	3,792,644.01	22,497	1,125,950.43	272,313	14,294,875.82	16,348	850,863.79
	Maximum (Product type)	**138,242**	**2,237,177.54**	**172,803**	**3,792,644.01**	**46,488**	**1,133,910**	**598,362**	**14,294,875.82**	**63,772**	**1,395,807.43**

Example - Create a Discontinuous Crosstab

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales for each product line by quarter and by order method. Since the report will have columns with data from different dimensions, you create a discontinuous crosstab report. Discontinuous crosstabs are also known as disconnected or disjoint crosstabs, or crosstabs with unrelated columns.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , drag the following data items to the crosstab:
   - **Product line** from **Product** to **Rows**
   - **Order method** from **Order Method** to **Columns**
   - **Revenue** from **Sales fact** to **Measures**
6. Pause the pointer over the query explorer button and click **Query1**.
7. On the **Toolbox** tab, drag **Data Item** to the **Data Items** pane.

   The **Data Item Expression** dialog box appears.
8. In the **Expression Definition** box, type the following and click **OK**:

   ```
   [Employee summary (query)].[Employee by region].[Region]
   ```
9. In the **Properties** pane, double-click the **Name** property, rename the data item **Region**, and press the Enter key.

10. Pause the pointer over the page explorer button and click **Page1**.
11. On the **Source Tab**, drag **Region** to the left of **Order method**.

    Tip: Ensure that the pointer is directly beside **Order method** before you drop **Region**. Otherwise, **Region** may appear as a nested row instead of a column.
12. Click **Region**.
13. In the **Properties** pane, double-click the **Sorting** property.
14. From the **Data Items** pane, drag **Region** to the **Sort List** pane.
15. Click the sort order button to sort quarters in ascending order.
16. Run the report.

Revenue	Americas	Asia Pacific	Central Europe	Northern Europe	Southern Europe	E-mail	Fax	Mail
Camping Equipment	481,445,781.04	421,639,391.62	343,645,848.36	180,851,396.88	161,454,246.13	75,899,094.63	23,054,398.48	21,348,644.09
Golf Equipment	217,262,995.22	193,677,873.68	153,632,833.39	84,424,300.9	77,413,364.7	47,933,933.16	15,241,303.27	12,693,287.48
Mountaineering Equipment	123,127,397.88	107,505,775.01	88,051,532.89	46,091,108.04	44,884,319.08	7,476,451.96	11,848,370.08	3,531,658.66
Outdoor Protection	23,002,647.68	19,716,018.32	17,488,870.77	8,346,431.17	7,440,328.31	5,882,477.87	1,966,484.72	2,098,391.71
Personal Accessories	132,249,058.98	116,715,219.51	1,540,675,699.15	49,825,913.97	46,207,416.17	42,651,086.54	17,962,985.46	6,419,357.03

Charts

You can use Report Studio to create many chart types, including column, bar, area, and line charts.

You can also create custom charts that combine these chart types.

Tip: To view the available chart types, from the **File** menu, click **New**, and then double-click the chart icon. You can also view the available chart types by adding a **Chart** object from the **Toolbox** tab in the **Insertable Objects** pane to an existing report.

We recommend that you work through the chart samples in the Try It Yourself appendix to familiarize yourself with some of the capabilities of charts in Report Studio. For more information, see "Try It Yourself - Create Charts" (p. 288).

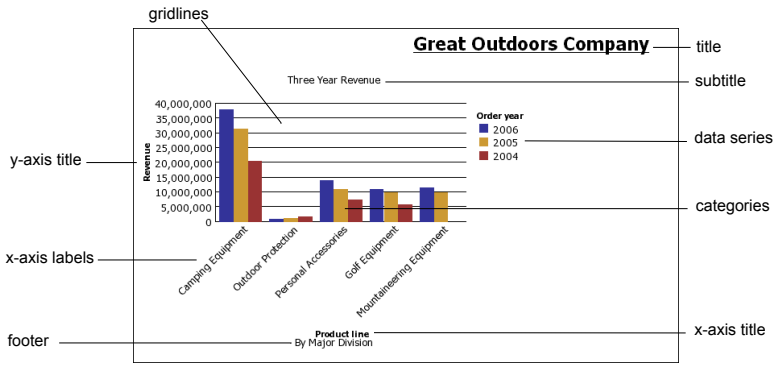
For information about different chart types, see "Chart Types" (p. 337).

You can also find information about how to create a chart in the Report Studio Tour.

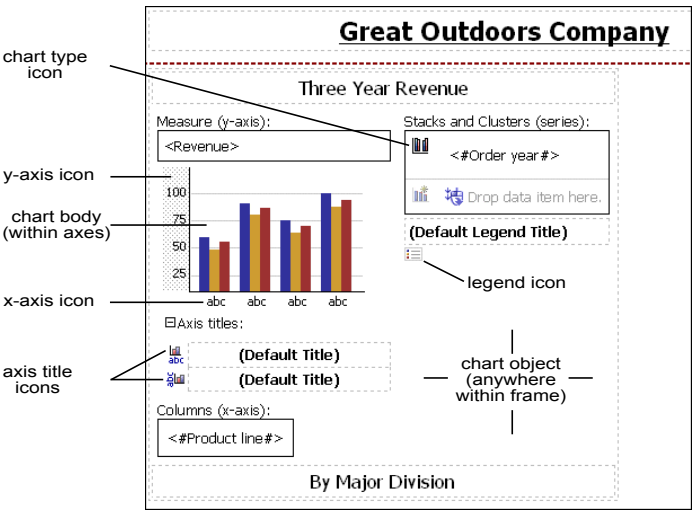
To create charts, you drag items from the **Insertable Objects** pane to the following drop zones:

- measures
- columns, bars, lines, areas, or points to represent data series
- groups, clusters, or stacks to represent categories of related data series

The following illustrates a typical chart as it appears in Cognos Viewer.



The following illustrates the same chart as it appears in the Report Studio interface. The y-axis icon is selected, and the **Axis titles** icon is expanded.



Example - Create a Column Chart to Plot Order Trends

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the contribution that each order method makes to revenue. You decide to create a column chart because it emphasizes the values of each order method for each year.

Steps

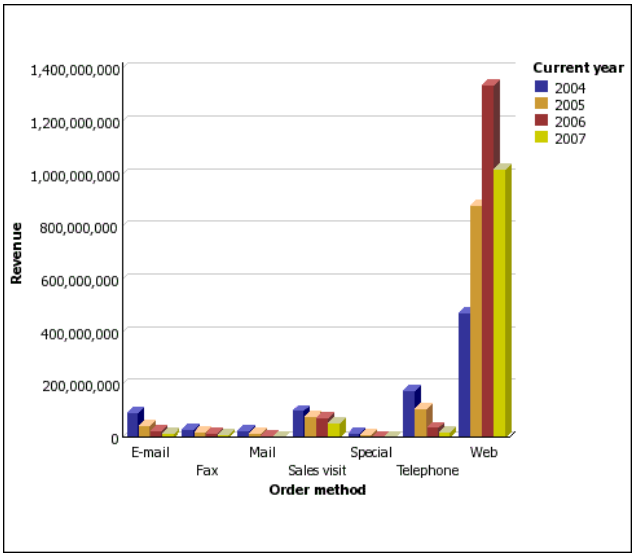
1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

   Report Studio starts.

3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Column**.
6. In the **Chart type** pane, click **Column with 3-D Visual Effect**.
7. Click **OK**.
8. In the **Insertable Objects** pane, on the **Source** tab, expand **Sales and Marketing (query)**, and **Sales (query)**.
9. Add the following data items:
   - From **Sales fact**, drag **Revenue** to the **Measure (y-axis)** drop zone.
   - From **Time Dimension**, drag **Current year** to the **Series** drop zone.
   - From **Order method**, drag **Order method** to the **Categories (x-axis)** drop zone.
10. Click the **Current year** icon in the **Series** drop zone, and then from the **Data** menu, click **Sort Ascending**.
11. Save the chart:
    - In the **Name** box, type

      Order Trends

    - Leave the default destination folder as **Public Folders**, and click **Save**.
12. Click the run button on the toolbar, and view the report.



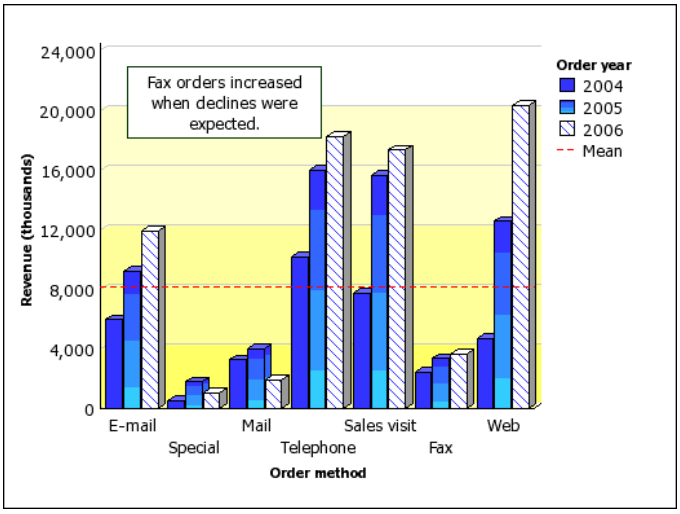
Customize a Chart

After you create a chart, you can customize it to suit your needs.

For example, the following chart was customized by:

- changing the palette
- changing the background
- adding a baseline
- adding a note
- formatting the axis values

You can make these and many other changes by changing the default properties of a chart.



You select the element in Report Studio to view its properties in the **Properties** pane. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the chart to change appearance or provide information in response to expressions or conditions.

When you change a chart property, you usually do not see the change until you run the report. Changes to the properties of labels and titles are reflected immediately.

These are some of the properties you can change in charts. These properties are available when you select the chart object, unless specified otherwise in the **Action to perform in the Properties pane** column.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, or footer	Under **Chart Titles**, change the **Title**, **Subtitle**, or **Footer** property.
Hide or show the legend, baselines (p. 102), markers, or notes (p. 107)	Under **Chart Annotations**, change the **Legend**, **Baselines**, **Markers**, or **Notes** property.
Hide or show the axes	Under **Axes**, change the **Y1 Axis**, **Y2 Axis**, or **Ordinal Axis** property.

Goal	Action to perform in the Properties pane
Hide or show the axis title or axis line	Select the y-axis or the ordinal axis. Under **General**, change the **Axis Line** or **Axis Title** property.
Hide or show the gridlines	Select the y-axis or the ordinal axis. Under **General**, change the **Gridlines** or **Minor Gridlines** property.
Hide or show the border around the legend	Select the legend icon. Under **General**, change the **Borders** property.
Hide or show the border around the chart object	Under **Box**, change the **Border** property.
Hide or show the tooltips **Note:** Some versions of Acrobat Reader do not support tooltips.	Under **Chart Labels**, change the **Tooltips** property.
Change y-axis properties, such as range, scale interval, and so on	Select the y-axis. Under **General**, change the **Minimum Value**, **Maximum Value**, **Scale Interval**, or **Scale** property.
Change the data format	Select the y-axis. Under **Data**, change the **Data Format** property.
Change ordinal axis properties, such as label truncation, skip interval, and so on	Select the ordinal axis. Under **General**, select the **Truncation**, **Allow Rotation**, **Allow Stagger**, or **Allow Skip** property.
Change the white space around the chart	Under **Box**, select the **Padding** or **Margin** property.
Change the color or pattern in the palette for columns, lines, and areas	Under **Color & Background**, select the **Palette** or **Conditional Palette** property.
Apply a palette to a series	Under **Color & Background**, select the **Series Color** property.
Change the default color or font (p. 101) for all chart elements	Under **Color & Background**, change the **Background Color**, **Foreground Color**, or **Fill Effects** property. Under **Font & Text**, change the **Font** or **Relative Alignment** property.

Goal	Action to perform in the Properties pane
Override the default font or color for axes and chart values	Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and change the **Font**, **Background Color**, **Foreground Color**, or **Fill Effects** property.
Resize the chart	Under **Positioning**, change the **Size & Overflow** property.
Change the 3-D appearance of a chart	Under **General**, change the **Depth** or **Visual Angle** property.
Insert a background image or watermark in the chart body, that is, the space between the axes	Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and change the **Background Image** property.
Insert a background image or watermark in the chart object	Under **Color & Background**, change the **Background Image** property.
Go to another report	Under **Data**, change the **Drill-Through Definitions** property.

Steps

1. Select the chart or chart element that you want to change:
   - To change general properties, such as size and color, click the chart object.
   - To change specific chart elements, such as a title or axis, click the element itself.

   Tip: To cancel a selection, press the Esc key.
2. In the **Properties** pane, click the property you want, and then make the desired changes.

   An ellipsis (...) button indicates that a dialog box provides further options.

   You may have to scroll to see all the properties.

Customize the Colors of a Chart

You can use the chart palette to control the colors of the columns, lines, or areas in a chart. For example, if you know that Telephone is the first in a data series of order methods and you want it to appear in blue, you use the palette to ensure that the first item in the series is blue.

You use a unique color, pattern, or gradient fill for each series in the chart. Patterns are useful when the report is printed on a black and white printer because colors do not print well in shades of gray.

Steps

1. Click the chart object.

2. In the **Color & Background** section of the **Properties** pane, click the ellipsis (...) button of the **Palette** property.
3. In the **Palette** dialog box, click the new palette entry button to define a new palette entry:
   - To apply color, click **Color**, specify the color properties, and click **OK**.
   - To apply a gradient, click **Gradient**, specify the **Direction**, **From color**, and **To color** properties, and click **OK**.
   - To apply a pattern, click **Pattern**, specify the **Pattern**, **Foreground color**, and **Background color** properties, and click **OK**.

     Tip: You can also select a predefined palette from the **Palettes** drop-down list.
4. Click **OK**.
5. Repeat steps 4 and 5 for each chart series.

Tips

- To delete a palette definition, select the definition in the **Palette** box and click the delete button .
- To change the order in which the colors, gradients, or patterns appear in the chart, use the arrow buttons under the **Palette** box to change their position.
- To copy and paste a palette, open the palette dialog box and press Ctrl+C to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and press Ctrl+V to paste the palette from the clipboard.

Example - Customize the Palette in the Order Trends Chart

You are a report author at The Great Outdoors Company. You have created a column chart that shows the contribution each order method makes to revenue.

You decide to give each column a distinctive look to better differentiate the values of each order method for each year.

Before you can try this example, you must create the chart in "Example - Create a Column Chart to Plot Order Trends" (p. 95).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Color & Background** section of the **Properties** pane, click the ellipsis (...) button of the **Palette** property.
4. Click the **Color** link.
5. On the **Web Safe Colors** tab of the **Color** dialog box, click a medium blue color.

6. Click **OK**.
7. Click the new palette entry button , and then click **Gradient**.
8. In the **Direction** list, click **Down**.
9. Click the **From color** link.
10. On the **Web Safe Colors** tab of the **From Color** dialog box, click a medium blue color and click **OK**.
11. Click the **To color** link.
12. On the **Web Safe Colors** tab of the **To Color** dialog box, click a lighter blue color and click **OK**.
13. Click the new palette entry button, and then click **Pattern**.
14. In the **Pattern** box, click a diagonal line pattern.
15. Click the **Foreground color** link.
16. On the **Web Safe Colors** tab of the **Foreground Color** dialog box, click a blue color and click **OK**.
17. Click **OK** again to close the **Palette** dialog box.

    Now you must add borders to the diagonal line pattern.
18. Click the chart type icon in the **Series** drop zone.
19. In the **General** section of the **Properties** pane, change **Borders** from **Hide** to **Show**.
20. Save the chart.

Change a Chart Background

You can use a solid color, a pattern, or a gradient fill effect to customize the chart background.

You can also use an image as a background for a chart. For more information, see "Customize a Chart" (p. 96).

Steps

1. Click the chart object.
2. Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and in the **Properties** pane, under **Color & Background**, click the ellipsis (...) button of the **Fill Effects** property.
3. In the **Effect** list, choose a color, a gradient fill effect, or a pattern:
   - To apply a color, click the **Color** link and specify the color properties.

- To apply a gradient fill effect, click **Gradient** and then specify the **Direction**, **From color**, and **To color** properties.
- To apply a pattern effect, click **Pattern** and then specify the **Pattern**, **Foreground Color**, and **Background Color** properties.

  The foreground color is the color of the selected pattern. The background color is the color of the area behind the pattern.

4. Click **OK**.

Tip: To remove a background fill effect, click **None**.

Example - Change the Background of the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to give your chart a visually interesting background fill.

Before you can try this example, you must create the chart in "Example - Customize the Palette in the Order Trends Chart" (p. 100).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and in the **Properties** pane, under **Color & Background**, click the ellipsis (...) button of the **Fill Effects** property.
4. In the **Effect** list, click **Gradient**.
5. Specify a gradient that goes from yellow to white:
   - In the **Direction** list box, click **Up**.
   - Click the **From color** link.
   - On the **Web Safe Colors** tab, click a light yellow color and click **OK**.
   - Leave white as the **To color**.
6. Click **OK**.
7. Save the chart.

Add a Baseline to a Chart

Baselines are horizontal or vertical lines which cut through the chart to indicate major divisions in the data. For example, you can add a baseline to show a sales quota or break-even point.

You can position the baseline using the following options.

Option	Position
Numeric Value	Uses a static numeric value.
Query Calculation	Uses a query calculation from the same query or from a different query. For more information, see "Create a Calculation" (p. 189).
Layout Calculation	Uses a layout calculation. For more information, see "Create a Calculation" (p. 189).
Member Calculation	Uses a position relative to a member. For example, a member calculation that uses an expression similar to [Query1].[Current year]=2006 places the baseline in the middle of the chart element that represents 2006. To place the baseline between 2006 and 2007 on the chart, that is, a half member width to the right, change the **Member Offset** property to 50%. To place the baseline between 2005 and 2006, change the **Member Offset** property to -50%. You can also type in 100, -200, and so on, to place the baseline where you want on the chart.
Statistical Minimum	Uses the following expression: (*25th percentile value*) - 1.5(*75th percentile value - 25th percentile value*) For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5. [(2.5) -1.5(5) = -5]
Statistical Maximum	Uses the following expression: (*75th percentile value*) + 1.5(*75th percentile value - 25th percentile value*) For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15. [(7.5) +1.5(5) = 15]
Mean	Uses the statistical mean, plus or minus a number of standard deviations, based on all charted data values on the specified axis.
Percentile	Uses a specified percentile.
Percent on Axis	Uses a percent of the maximum axis value.

If you apply more than one baseline, you can specify in which order they should be drawn when the report runs. They are drawn from top to bottom. If you have two baselines with the same coordinates, the first one in the list is drawn first and the next one is drawn on top.

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Baselines**, and then click the ellipsis (...) button.
3. Click the new button .
4. Click the desired type in the list and provide any necessary criteria, such as a value or percentage.
5. If you add more than one baseline, specify their order using the up and down arrows.
6. Click **OK**.

   A baseline icon appears in the **Markers, Notes, and Baselines** box.
7. To define the line style, click the baseline icon .
8. To define the text style, click the baseline text icon.
9. To add more baselines, repeat steps 1 to 6.

Tip: To delete a baseline, click its baseline icon and click the delete button .

Example - Add a Baseline to the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to add a baseline to indicate the mean revenue for the Order Trends chart.

Before you can try this example, you must create the chart in "Example - Change the Background of the Order Trends Chart" (p. 102).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Chart Annotations** section of the **Properties** pane, click **Baselines**, and then click the ellipsis (...) button.
4. Click the new button .
5. Click the **Data Mean** type in the list, leave **Distance From Mean** at zero, and click **OK** twice.

   A baseline icon appears in the **Markers, Notes, and Baselines** box.
6. Click the baseline icon to define the line style.

7. In the **General** section of the **Properties** pane, change the **Line Color** to **Red**.
8. In the **General** section of the **Properties** pane, change the **Line Style** to **Dash**.
9. Save the chart.

Customize the Lines in a Chart

You can change the color, style, and weight of the lines in a chart. You can apply these changes to the x-axis and y-axis lines, as well as to the major and minor gridlines.

You can also

- specify whether to use a linear or a logarithmic scale for the y-axis
- specify a scale interval for the y-axis
- show or hide major and minor gridlines
- show regression lines on scatter charts (p. 343) and bubble charts (p. 344)

Regression Lines

Regression lines graphically illustrate trends in data series. Also called trend lines, they are commonly used when charting predictions. A regression line is typically a line or curve that connects two or more points in the series and then extends into the future.

You can specify the following types of regression lines to determine the position and slope of the line.

Steps

1. Click the y-axis icon or the x-axis icon of the chart.
2. In the **General** section of the **Properties** pane, click **Axis Line**, **Gridlines**, or **Minor Gridlines**, then click the ellipsis (...) button.
3. Click the color, weight, and style that you want.
4. Click **OK**.

Tips

- To specify a linear or logarithmic scale for the y-axis, in the **General** section of the **Properties** pane, change the **Scale** property.
- To specify a scale interval for the y-axis, in the **General** section of the **Properties** pane, change the **Scale Interval** property.
- To show minor gridlines, in the **General** section of the **Properties** pane, click **Minor Gridlines**, click the ellipsis (...) button, select the **Show minor gridlines** check box, and type a number under **Number of minor gridlines**.

- To show regression lines, click a scatter chart or bubble chart object and, in the **Chart Annotations** section of the **Properties** pane, change the **Regression Line** property to **Show**.

  When you show regression lines, you must then select their properties. Click the regression line icon in the report layout. In the **General** section of the **Properties** pane, change the **Line Styles**, **Regression Type**, and **Number of Regression Lines** properties.

Add a Marker to a Chart

Markers are notations along the axis of a chart that designate a point on the scale that may help you analyze or understand the data. The numeric position of the marker can be based on

You can position the marker using the following options.

Numeric Position Type	Position
Numeric Value	Uses a static numeric value.
Query Calculation	Uses a query calculation from the same query or from a different query. For more information, see "Create a Calculation" (p. 189).
Layout Calculation	Uses a layout calculation. For more information, see "Create a Calculation" (p. 189).
Statistical Minimum	Uses the lowest value on the axis.
Statistical Maximum	Uses the highest value on the axis.
Mean	Uses the statistical mean, plus or minus a number of standard deviations, based on all charted data values on the specified axis.
Percentile	Uses a specified percentile.
Percent on Axis	Uses a percent of the maximum axis value.

If you apply more than one marker, you can specify in which order they should be drawn when the report runs. They are drawn from top to bottom. If you have two markers with the same coordinates, the first one in the list is drawn first and the next one is drawn on top.

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Markers**, and then click the ellipsis (...) button.

3. Click the new button and specify the **Numeric position type**, **Marker label**, **Marker shape**, **Numeric value**, and **Color** properties.
4. If you add more than one marker, specify their order using the up and down arrows.
5. Click **OK** twice.

   The new marker appears in the **Markers, notes, and baselines** box.

   Tip: To delete a marker, click the marker icon and text, and then click the delete button.
6. Run the chart to view the marker.

Add a Note to a Chart

Notes are pieces of text that appear in the chart. You can add notes to a chart to provide additional detail, such as explanatory information, or to highlight a notable value.

You can use text, an item label or value, or a report expression as the source for a note.

Notes overwrite whatever is under them so position them properly.

If you apply more than one note, ensure that each note has a different position in the report so that they don't overwrite each other. You can also specify in which order they should be drawn when the report runs. They are drawn from top to bottom. If you have two notes with the same coordinates, the first one in the list is drawn first and the next one is drawn on top.

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, and then click the ellipsis (...) button.
3. Click the new button and click **OK**.

   A note icon appears in the **Markers, notes, and baselines** box.
4. Click the note icon to define size, position, and note border in the **Properties** pane.

   The location and size are statically set based on the number of pixels.
5. Double-click the note text icon to define the text.

   Tip: To delete a note, click the note icon, and then click the delete button.

Example - Add a Note to the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to add a note to draw attention to an unexpected result.

Before you can try this example, you must create the chart in "Example - Add a Baseline to the Order Trends Chart" (p. 104).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, then click the ellipsis (...) button.

   The **Notes** dialog box appears.
4. Click the new button , and then click **OK**.

   A note icon appears in the **Markers, notes, and baselines** section.
5. Click the note icon .
6. In the **General** section of the **Properties** pane, change **Bottom Position** to 285 and **Left Position** to 110.

   The location and size are statically set based on the number of pixels.
7. In the **General** section of the **Properties** pane, click **Note Border**, then click the ellipsis (...) button.
8. Select the **Note border** check box, click a line color and line style, and then click **OK**.
9. Run the report to view the note.

   If necessary, change the position again.
10. Double-click the note text icon and type

    Fax orders increased when declines were expected.
11. Save the chart.

Example - Format the Axis Values of the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to format the y-axis values to make them easier to read.

Before you can try this example, you must create the chart in "Example - Add a Note to the Order Trends Chart" (p. 107).

Steps

1. Open the **Order Trends** chart.
2. Click the y-axis icon.

3. In the **Data** section of the **Properties** pane, click **Data Format**, and then click the ellipsis (...) button.
4. Under **Format type**, click **Number**.
5. Under **Properties**, click **Scale**, in the drop-down list click **-3**, and click **OK**.
6. Expand **Axis titles**.
7. Double click the y-axis title icon and type

   Revenue (thousands)
8. Save the chart.

Example - Create a Drill-through Chart

You create a drill-through report to link two reports containing related information. You can then access related or more detailed information in one report by selecting a value in the chart.

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a chart that shows the revenue for each product line and allows the reader to drill through from the revenue chart to view the product details for any item selected. You create two reports, a target list report that contains the details for the item, and a source chart that shows the product line revenue.

For more information about using drill-though reporting in Report Studio, see "Set Up Drill-through Access in a Report" (p. 273).

Steps to Create the Target Report

1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

   Report Studio starts.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)** and **Sales (query)**.
6. Add the following data items to the list by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**
   - **Introduction date**
   - **Product image**

Tip: You can find these data items in the **Product** folder.

Now you must create a filter to use as a drill-through parameter. A drill-through parameter begins and ends with a question mark (?) symbol.

7. Click the filters button.
8. Click the add button and type the following in the **Expression Definition** window:

   [Sales (query)].[Product].[Product line]=?p_PL?
9. Click **OK**.
10. Save the report as **Product Line Details**.

Steps to Create the Source Chart

1. Create a new report.
2. In the **New** dialog box, click **Chart** and click **OK**.
3. In the **Chart group** pane, click **Column**.
4. In the **Chart type** pane, click **Column**.
5. Click **OK**.
6. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)** and then **Sales (query)**.
7. Expand **Sales fact** and drag **Revenue** to the **Measure (y-axis)** drop zone.
8. Expand **Order method** and drag **Order method** to the **Series** drop zone.
9. Expand **Product** and drag **Product Line** to the **Categories (x-axis)** drop zone.
10. From the **Data** menu, click **Drill Behavior**.
11. In the **Basic** tab, in the **Report Drill Capabilities**, select the **Allow package based drill-through** check box.
12. Click **OK**.
13. Right-click the chart object and click **Drill-Through Definitions**.
14. Click the **New Drill Through** button.
15. Under **Report**, click the ellipsis (...) button.
16. Select the **Product Line Details** report you created previously and click **Open**.
17. Under **Action**, select **Run the Report**.
18. Under **Format**, select **HTML**.
19. Click the edit button.

Any existing drill-through parameters appear. You see the parameter you created for **Product Line Details**.

20. For item **p_PL**, under **Method**, click **Pass data item value**, and under **Value**, click **Product line**.
21. Click **OK** twice.
22. Save the chart as **Product Revenue**.
23. Click **Run**.

When the report is run, the chart will show the product lines as clickable links. When a product line is clicked, the second report will be run for that product line.

Specify the Chart Label Text and Value Text Shown

You can specify which labels and values to use when generating the text shown on the chart.

When you show all the labels and values on some chart types, such as scatter charts, bubble charts, and polar charts, the text shown may be too long. For example, a bubble chart may show the values and the labels for the series, the categories, the x and y measures, and the bubble size.

Steps

1. Click the chart object.
2. In the **Chart Labels** section of the **Properties** pane, click **Values**, and then click the ellipsis (...) button.
3. Click the labels and values that you want to show.
4. Click **OK**.

Specify Custom Label Text

You can override the default label properties.

By default, chart labels use the name of the underlying data item label. You can change the label text to make it more meaningful. For example, you can rename a chart item labeled ISO_3_Letter_Code to Country.

Steps

1. Click the chart node member or measure.
2. In the **Chart Labels** section of the **Properties** pane, change the **Custom Label** property from **Hide** to **Show**.

   A new chart text item appears.
3. Do one of the following:
   - To change the text of the label, double click the chart text item and type in the text you want.

- To change the source of the label text, select the new chart text item. In the **Text Source** section of the **Properties** pane, click the source type you want to use in the **Source Type** drop-down list.

If nothing is specified in the new chart text item, the label appears blank.

Tip: To remove a custom label, change the **Custom Label** property from **Show** to **Hide**.

Specify the Properties of a Gauge Chart

The default gauge chart uses a band divided into thirds that runs from green to red. You can change the properties to improve the presentation of your data. For example, you may want change the colors in the band, or you may want to use a number instead of a percentage as a threshold.

Steps

1. Open the gauge chart that you want.
2. Click the chart object.
3. In the **Color & Background** section of the **Properties** pane, click the ellipsis (...) button of the **Gauge Palette** property.
4. Specify the following properties:
   - To change the color of a palette entry, under **Palette** select the color, click **Color**, specify the color properties, and click **OK**.
   - To change the boundary value of a palette entry, under **Palette** select the boundary value and specify the color properties, and click **OK**
   - To define a new palette entry, click the new palette entry button .
   - To choose a threshold style, click **Palette**, and click a style in the drop down list.
5. Click **OK**.

Tips

- You can choose whether to use discrete colors or continuous colors.
- To use a number instead of a percentage as a boundary, click a number under **Palette**, clear the **Percentage** check box, and then type a number in the **Numeric boundary** box.
- To copy and paste a gauge palette, open the gauge palette dialog box and click the copy icon to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and click the paste icon to paste the palette from the clipboard.

Create a Drill-up and Drill-down Chart

If you use a dimensionally modeled data source, you can create a chart that allows you to drill down to lower-level data or drill up to higher-level data.

Drilling up and down allows you to view more general or more detailed information on your data within a predefined dimensional hierarchy.

This is an example of a dimensional hierarchy:

Years - Year - Quarter - Month

Before you begin, ensure that you are using a dimensionally modeled data source.

For more information about using drill-up and drill-down reporting in Report Studio, see "Create a Drill-up/Drill-down Report" (p. 279).

Steps

1. Open a chart that uses a dimensionally modeled data source.
2. From the **Data** menu, select **Drill Behavior**.
3. On the **Basic** tab, in the **Report Drill Capabilities,** select the **Allow drill up and down** check box.

   By default, the system determines which items can be drilled on, based on the dimensional structure.

   On the **Basic** tab, you can make drilling unavailable for any data item by selecting the parameter in either the **Disable drill up for** box or the **Disable drill down for** box.

   On the **Advanced** tab, you can change the drill-up or drill-down behavior for any parameter by selecting the parameter and then choosing one of the desired behaviors.
4. Click **OK**.

The chart generates links for any item that can be drilled down on.

You can drill down or drill up by right clicking and choosing the action from the context menu. The menu items are unavailable if an item cannot be drilled up or down on.

Example - Show Values in the Chart Legend

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the quantity of items sold for each product line by order method. To show how much the quantity represents in revenue, you want to show the revenue for each order method in the legend.

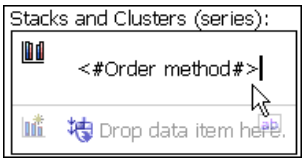
Steps

1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

   Report Studio starts.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Column**.

6. In the **Chart type** pane, click **Column**.
7. Click **OK**.
8. In the **Insertable Objects** pane, on the **Source** tab, expand **Sales and Marketing (query)** and **Sales (query)**.
9. Expand **Sales fact** and drag **Quantity** to the **Measure (y-axis)** drop zone.
10. Expand **Order method** and drag **Order method** to the **Series** drop zone.
11. Expand **Product** and drag **Product Line** to the **Categories (x-axis)** drop zone.
12. From the **Structure** menu, clear **Lock Page Objects**.

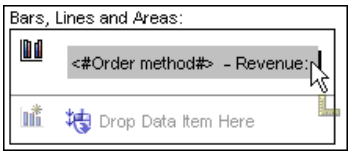
    If you do not clear **Lock Page Objects**, you will be unable to insert a text item in the legend.
13. In the **Insertable Objects** pane, on the **Toolbox** tab, drag a text item into the legend next to **Order method**.



14. In the **Text** box, type

    - Revenue:

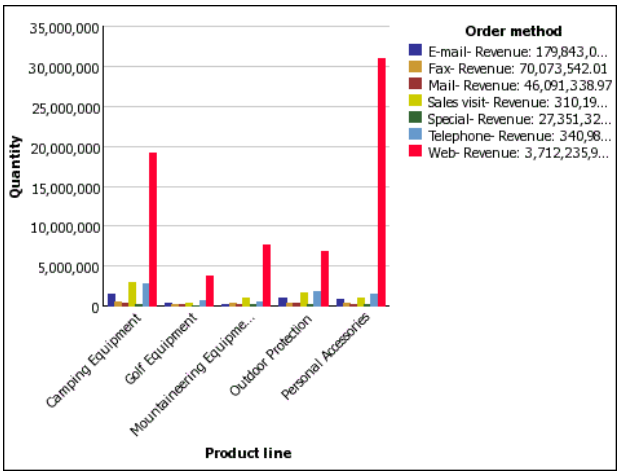
    You must insert a trailing space after the colon.
15. In the **Insertable Objects** pane, on the **Source** tab, drag **Revenue** to the right of the text item.



16. Click the run button on the toolbar, and view the report.

The revenue is listed beside each legend item.

Tip: To reposition the legend in the chart, click the legend icon, change the **Absolute Position** property to **Yes**, and specify the **Left Position** and **Top Position** values.



Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels

You want to use a revenue expression as a chart legend title. If you get an error message saying that the query context of a layout object cannot be determined, you must define the property list for the item it refers to. You must first add the desired data item to the query before you can define its property list.

For more information, see "Specify the List of Data Items for an Object" (p. 197).

Steps

1. Open the chart that you want to customize.
2. Click the query explorer button .
3. Click the query icon.
4. In the **Insertable Objects** pane, drag the desired item to the **Data Items** window to add it to the query.
5. Click the page explorer button to return to the chart.
6. In the **Insertable Objects** pane, drag the desired item to the layout object.
7. In the **Properties** pane, under **Data**, click the ellipsis (...) button to open the **Properties** dialog box.
8. Select the data item to define.
9. Click **OK**.

Example - Customize the Axis Titles

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the total revenue for the report in the horizontal axis title.

Charts contain several titles, such as axis titles, report titles and subtitles, and the legend title.

By default, the axis titles are managed for you. To customize an axis title, you drag text items or data items to the axis title area in the chart. You can use combinations of text, data items, and report expressions in titles.

Steps

1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

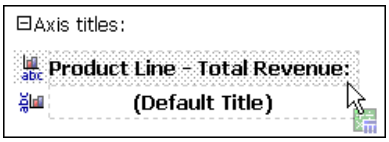
   Report Studio starts.
3. In the **Welcome** dialog box, click **Create a new report or template**.

   In the **New** dialog box, click **Chart** and click **OK**.
4. In the **Chart group** pane, click **Column**.
5. In the **Chart type** pane, click **Column**.
6. Click **OK**.
7. In the **Insertable Objects** pane, on the **Source** tab, expand **Sales and Marketing (query)** and **Sales (query)**.
8. Expand **Sales fact** and drag **Revenue** to the **Measure (y-axis)** drop zone.
9. Expand **Order method** and drag **Order method** to the **Series** drop zone.
10. Expand **Products** and drag **Product line** to the **Categories (x-axis)** drop zone.
11. Expand **Axis titles**.
12. Click the horizontal axis title icon.

    Tip: You can use the **Properties** pane to change the axis title properties, such as font & text, positioning, color & background, and so on.
13. In the **Insertable Objects** pane, on the **Toolbox** tab, drag a text item into the horizontal axis text.
14. In the **Text** box, type

    Product Line - Total Revenue:

    You must insert a trailing space after the colon.
15. In the **Insertable Objects** pane, on the **Toolbox** tab, drag a query calculation to the right of the text.



16. In the **Create Calculation** dialog box, type

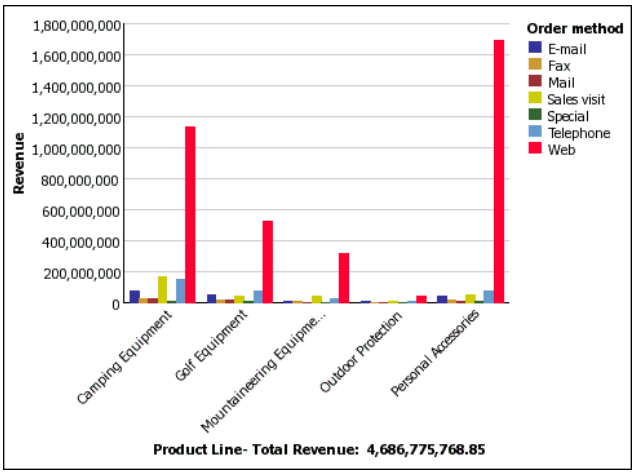
Total Revenue for Report

17. Click **OK**.
18. In the **Expression Definition** box, type the expression:

    total (**[Revenue] for report**)
19. Click **OK**.
20. Click the run button on the toolbar, and view the report.

The title shows the calculated total revenue for **Product line**.

Tip: You can use conditional formatting (p. 254) to conditionally style the titles.



Example - Create a Conditional Palette

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows one pie slice for each order method. You also want to highlight the pie slice for the order method that has the highest quantity.

You create a condition that shows you which order method has sold a quantity less than 1,000,000.

Steps

1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

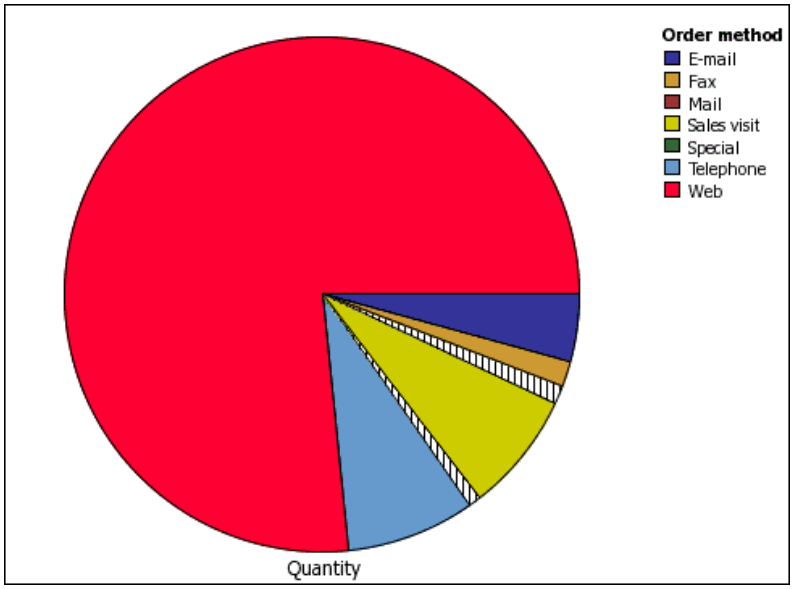
   Report Studio starts.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Pie, Donut**.
6. In the **Chart type** pane, click **Pie**.
7. Click **OK**.

8. In the **Insertable Objects** pane, on the **Source** tab, expand **Sales and Marketing (query)**, **Sales (query)** and **Sales fact**.
9. Drag **Quantity** to the **Default measure** drop zone.
10. Expand **Order method** and drag **Order method** to the **Pie Slices** drop zone.

    You want to create a condition that shows you which order method has sold a quantity less than 1,000,000.
11. Select the pie chart.
12. In the **Color & Background** section of the **Properties** pane, open the **Conditional Palette** properties dialog box.
13. Under **Variable**, select **<New Boolean Variable>**.
14. In the **New Variable** dialog box, type

    LowQuantity
15. In the **Expression Definition** box, type the expression:

    [Query1].[Quantity] < 1000000
16. Click **OK** to close the expression editor. Leave the **Conditional Palette** dialog box open.
17. Under **Effect**, click **Pattern**.
18. Under **Pattern**, click a pattern.
19. Click **OK**.
20. In the **Box** section of the **Properties** pane, change the **Borders** property from **Hide** to **Show**.
21. Run the report.

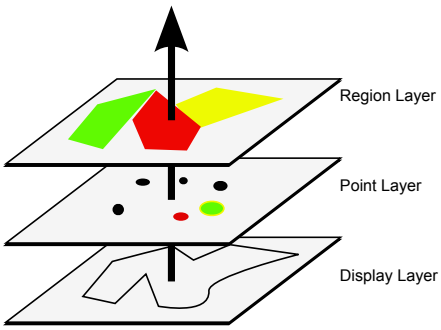


Maps

Report Studio provides a set of maps that you can use to represent tabular data in a spatial context. For example, on a map of the world, countries can be colored to represent the level of revenue. To edit the maps or create additional maps, use Map Manager. For more information, see "Edit a Map" (p. 129).

Maps are most often used to show geographical areas, but they can be used to show other spatial information, such as a floor plan in a building, seats in an airplane, or parts of the human body.

A map in Cognos 8 consists of a collection of layers. Each layer contains different information and acts like a transparency that is placed on top of the map. Each layer adds more information to the map. For example a map of the world may contain information related to countries on one layer and information related to cities on another level.



Report Studio maps provide the following three types of layers:

- Region layer

  Specifies the regions on a map to be differentiated according to values in the data source. For example, to show the revenue level for each country on a map of the world, choose Country as the region layer and then specify that the color of each country is based on the revenue value for that country. Regions can be set up for drilling through to other reports.

- Point layer

  Specifies the points to be placed on a map. The color and size of the points is based on the data that you select. For example, you choose to show cities as points on a map and set the color of each point by revenue and the size of each point by profit. Points can be set up for drilling through to other reports.

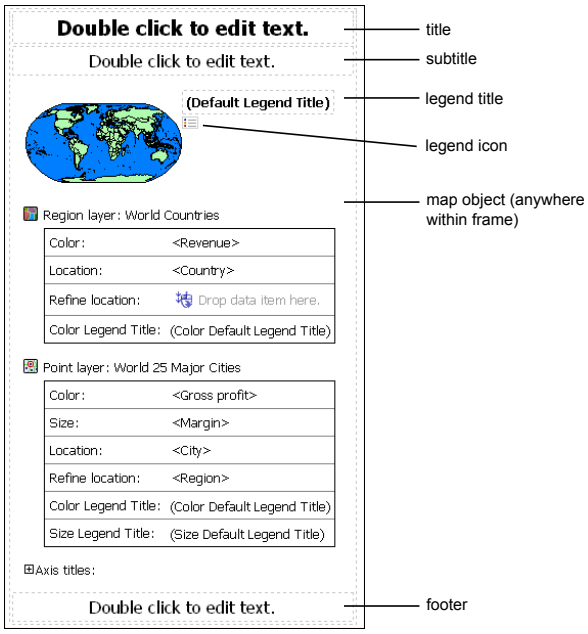
- Display layer

  You can show or hide items such as grid lines or capital cities. This layer is determined in the map file and not in the data source.

Note: If you intend to create CSV or XML output from your map, use only a point layer or a region layer. CSV and XML do not support the simultaneous use of both layers in the same map. Only one layer will be rendered in the output.

Parts of a Map Report

The following shows the parts of a map as they appear in the Report Studio interface.



Example - Create a Map Report

You are a report author at The Great Outdoors Company. You are asked to show how revenue for the company is distributed throughout the world. This information can be shown in tabular format using a list report, but a map will create a more meaningful presentation. You decide to create a report that contains a map of the world showing the distribution of revenue by country.

Steps

1. In Cognos Connection, go to the **GO Data Warehouse (query)** package.
2. Click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.

   Report Studio starts.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Map** and click **OK**.
5. In the **Choose Map** dialog box, in the **Maps** pane, expand the folder **World** and click **World**.
6. In the **Region Layers** box, click **Countries + Territories**.
7. In the **Point Layers** box, click **None**.

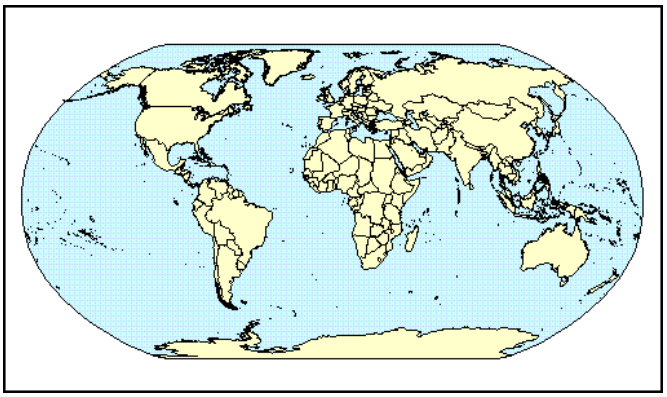
8. In the **Display Layers** box, click **Oceans**.

   Tip: You can select multiple display layers or deselect a display layer by pressing Ctrl+right-click.

9. Click **OK**.

   Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map background.

10. Click the run button on the toolbar, and view the map.

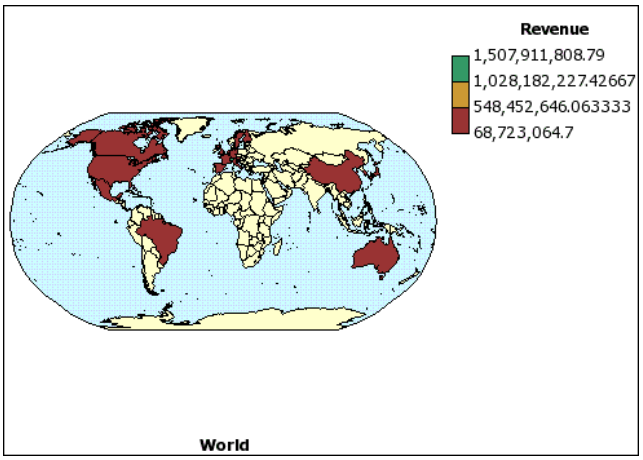


Example - Define Data Values for the Region Layer

The map that you created in the previous topic is not yet linked to a data source. You will now specify the data values from your data source that will determine the color of each region.

Steps

1. In the **Insertable Objects** pane, expand **Sales and Marketing (query)**, **Sales (query)** and **Sales fact**.
2. Drag **Revenue** to the **Color** drop zone.
3. Expand **Employee by organization**.
4. Drag **Country** to the **Location** drop zone.
5. Select the map object by clicking the map background.
6. Run the report.



Ignore Data with no Features

You can specify that you do not require a match for each item in the data source. The report can then run successfully even if the map does not contain a match for each item in the data source for a particular object.

For example, an error message appears and the report does not run if you are using cities in the point layer and a city in the data source does not have the same name in the map file. If you ignore data with no features, then the data that has a match in the map file will be plotted and the data that does not have a match will be omitted.

Steps

1. Click the background of the map.
2. In the **Data** section of the **Properties** pane, click the **Ignore Data with No Features** property.
3. Set the value to **Yes**.

Match Data Values to Names in the Map File

If the **Ignore Data with No Features** property is set to **No**, then each object that is called from the data source must have a matching label in the specified layer of the map file. For example, if your data source has a country named United States and the layer in the map file labels the same country USA, then there is a mismatch that must be corrected. Report Studio only makes you aware of a mismatch if each object in your data source does not have a corresponding label in the map file. If there are extra labels in the map file that do not have a match in the data source, the report will run without an error message.

A mismatch between your data and the map file must be corrected by the report author. It can not be corrected by a consumer of the map report at run time. There are two ways to correct a mismatch between your data and the labels in the map files. You can use Map Manager to edit the labels in the layers of the map file, or you can use the dictionary property to create an alias for each mismatched object. When you use the dictionary property, it resolves the mismatch only for a single report, and is not shared with other reports. If you intend to continue using a map with the same

data source, it is best to edit the map in Map Manager so that the labels match the objects in your data source.

For information about using Map Manager, see the Map Manager *Installation and User Guide*.

Steps to Create an Alias

1. Select the map object.

   The Title Bar of the **Properties** pane now shows the word **Map**.

2. In the **General** section of the **Properties** pane, click **Dictionary**, then click the ellipsis (...) button.

3. Click the new button .

4. In the **Dictionary Entry** dialog box, click **Search**.

5. In the **Search string** box, type a word or part of a word that you want to search the map file for.

   For example, if you are searching for United States, type in part or all of the name.

6. In the **Search map layer** box, click the layer that you want to search, and click **Search**.

7. In the **Matching features** box, click the label that you want to match your data source to and click **OK**.

8. In the **Alias** box, type the name as it appears in the data source and click **OK**.

   For example, if the country in your data source is named USA, type USA as the alias.

   To find out the name for the objects in your data source, run a list report. For example, you can run a list report to show the names of all the countries in your data source. For more information see "List Reports" (p. 79).

9. Click **OK**.

Define Data Values for the Point Layer

The Point layer in a map is used to visually represent data for point locations, such as cities or sales outlets. Both the color and size of points can be based on data from your data source.

Steps to Set the Color and Size of Points

1. Open the **Choose Map** dialog box:

   - When you create a new map report, this dialog box appears automatically.
   - If you are already in a map report, double-click the map background.

2. In the **Point Layers** box, select the layer containing the points that you want shown on the map.

   For example, on a map of the world you may want the points to represent cities.

3. Click **OK**.

4. In the **Insertable Objects** pane, drag an object to the **Color** drop zone in the **Point Layer**.

   For example, to have the color of the point based on revenue, drag **Revenue** from the **Insertable Objects** pane to the **Color** drop zone.

5. In the **Insertable Objects** pane, drag an object to the **Size** drop zone in the **Point Layer**.

Steps to Set the Location of Points

1. In the **Insertable Objects** pane, drag an object to the **Location** drop zone in the **Point Layer**.

   The object must be supported in the map file as a point location. For example, in the World sample map, city is supported as a point location, but country is not.

2. If you need to refine the location, drag an object to the **Refine Location** drop zone.

   Use this drop zone when there is more than one location with the same name. For example, if you try to run a report with cities in the point layer, and there is more than one city in your data source with the same name, the report does not run. An error message indicates that there are cities in your data source with duplicate names. You can differentiate the cities by using the data object **Region** to refine the location.

Add Another Color to the Region or Point Layer

You can add colors for regions or points and specify values to determine when those colors are shown.

Steps

1. In the report, click the **Region Layer** or **Point Layer**.
2. In the **Color & Background** section of the **Properties** pane, click **Palette**, and then click the ellipsis (...) button.
3. Click the new button , and click **Color**.
4. A new color is added to the list of colors.
5. With the new color selected, click **Color** in the right pane of the dialog box, and select a color.
6. Change the percentage boundaries for the colors.

   Tip: To specify absolute values rather than percentages, clear the **Percentage** check box.

7. Click **OK**.

Specify the Size of Points

The size of the points used on a map is defined in the **Properties** pane.

Steps

1. In the point layer, click the measure in the **Size** drop zone.

For example, if the measure is Revenue, click it.

2. In the **General** section of the **Properties** pane, click **Minimum Size**, and select or type the minimum point size for points on the map.
3. Click **Maximum Size**, and select or type the maximum point size for points on the map.

Add Titles, Subtitles, Footers, and Axis Title

You can add map titles, subtitles, and footers. In addition, the axis title is shown below the map. By default, the axis title shows the name of the map specified in the map file.

Steps

1. Select the map object.
2. In the **Chart Titles** section of the **Properties** pane, click the **Title, Subtitle, Footer,** or **Axis Title** property and click **Show**.
3. Double-click in the box that appears on the report and type the text that you want.

   By default, the axis title uses the name of the map.

Add Legend Titles

There are legend titles for the entire legend, for the color of the regions, for the color of the points, and for the size of the points.

Steps to Change the Legend Title for the Entire Map

1. If the legend title is not showing, click the legend icon . In the **General** section of the **Properties** pane, set the **Legend Title** to **Show**.
2. In the report, double-click the legend title and type the title you want.

Steps to Change the Legend Title for the Region Color, Point Color, or Point Size

- By default the legend titles are taken from the object you've selected from the data source. To change a title, do one of the following:
  - In the **Insertable Objects** pane, drag a text or calculation object to the **Color Legend Title** drop zone in the Region Layer or Point Layer, or the **Size Legend Title** drop zone in the Point Layer.
  - Double-click the **Color Legend Title** or **Size Legend Title** drop zone, to change from the default legend title, then double-click the drop zone again. In the **Text** dialog box, type text for the legend title.

Add a Note to a Map

You can add one or more notes, determine their position in the map report, and specify borders around notes.

Steps to Add a Note

1. Select the map object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, and then click the ellipsis (...) button.
3. Click the new button , and click **OK**.

   A note icon with the words **New Note** appears in the report.
4. Click **New Note** next to the note icon .
5. In the **Text Source** section of the **Properties** pane, click the **Text** property, and then click the ellipsis (...) button.
6. Type the text that you want to appear in the note and click **OK**.

Steps to Position a Note and Add a Border

1. In the report, click the note icon .
2. In the **Positioning** section of the **Properties** pane, type values to specify the bottom position, left position, height, and width for the note.

   The location of the note is defined by the number of pixels.
3. Use the **Note Border** property to specify a border around the note.
4. Run the report to view the note.

   If necessary, change the position again.

Customize a Map

When you select an element in a map, you can view its properties in the **Properties** pane. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the map to change appearance or provide information in response to expressions or conditions.

When you change a map property, you usually do not see the change until you run the report. Changes to the properties of labels and titles are reflected immediately.

These are some of the properties you can change in maps. These properties are available when you select the map object, unless specified otherwise in the **Action to perform in the Properties pane** column.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, footer, or axis title	Under **Chart Titles**, change the **Title**, **Subtitle**, **Footer**, or **Axis Title** property.
Hide or show the legend	Under **Chart Annotations**, change the **Legend** property.
Hide or show map labels	Select the region or point layer. Under **Chart Labels**, change the **Labels** property.
Hide or show values on the map	Select the region or points layer. Under **Chart Labels**, change the **Values** property.
Hide or show the border around the legend	Select the legend icon. Under **Box**, change the **Borders** property.
Change the border around the map object	Under **Box**, change the **Border** property.
Hide or show the tooltips **Note:** Some versions of Acrobat Reader do not support tooltips.	Under **Chart Labels**, change the **Tooltips** property.
Change the amount of white space around the map	Under **Box**, change the **Padding** or **Margin** property.
Change the default colors for all map elements	Under **Color & Background**, change the **Background Color**, **Foreground Color**, or **Fill Effects** property.
Change the font and the alignment of text	Under **Font & Text**, change the **Font** or **Relative Alignment** property.
Resize the map	Under **Positioning**, change the **Size & Overflow** property.
Change the font for the legend	Select the legend icon. Under **Font & Text**, change the **Font** property.
Change the format of values in the legend	Select the value in the region or point layer. Under **Data**, change the **Data Format** property.

Steps

1. Select the map object or map element that you want to change:
    - To change general properties, such as size and color, click the map object.

- To change specific map elements, such as a layer or title, click the element itself.

Tip: To cancel a selection, press the Esc key.

2. In the **Properties** pane, click the property you want, and then make the desired changes.

   An ellipsis (...) button indicates that a dialog box provides further options.

   You may have to scroll to see all the properties.

Drill Through to Another Report From a Map

You can link regions or points on a map to another report. For example, on a map of the world, you can specify that when you click on China, a map of China opens.

Steps to Set Up a Filter in the Target Report

1. Open the target report.
2. From the **Data** menu, click **Filters**.
3. On the **Detail Filters** tab, click the add button .
4. In the **Available Components** box, click the source or **Data Items** tab to select the data item you want to use.

   For example, if you want the target report to open when Canada is clicked in the source report, expand **Countries** and double-click **Country**.
5. In the **Expression Definition** box, type an operator after the data item or select an operator from the **Functions** tab and then enter a value.

   For example, if you want the report to open when Canada is clicked in the source report, the expression would be as follows:

   [Country]='Canada', where [Country] is the name of the data item in the package.
6. Save the target report.

Steps to Set Up a Drill-Through Link in the Source Report

1. Open the source report.
2. Select the region layer or the point layer.
3. In the **Data** section of the **Properties** pane, click **Map Drills**, and then click the ellipsis (...) button.
4. In the **Map Drills** dialog box, click the new button.
5. In the **Search String** box, type the name of the feature you want to use for the drill-through link, select it from **Matching Features** box, then click **OK**.

   For example, if you want a report to open when you click on Canada, search on all or part of the word **Canada** then select **Canada**.

6. In the **Drill-Through Definitions** dialog box, click the new button .
7. Click the ellipsis (...) button beside the **Report** box, and select the target report.
8. Click **OK**.
9. Run the report.

When you click the selected feature in the source report, the target report will open.

Edit a Map

Administrators and modelers use a Windows utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

With Map Manager you can edit the labels in maps to be consistent with the object names in your data base.

For instructions, see the Map Manager *Installation and User Guide*.

Additional Maps

Additional maps are available from the Cognos Global Customer Services Web site. Much of the mapping data on the Web site is derived from MapInfo Corporation, the Cognos preferred provider for location intelligence. MapInfo provides a wide variety of global data including

- detailed accurate roads
- political, postal, and census boundaries
- industry-specific data, such as communication system boundaries, insurance risk related data, and business prospect data

The Cognos Global Customer Services Web site provides a list of data providers and descriptions of the map data that Cognos provides as part of Cognos 8.

Location Intelligence

Sophisticated mapping functionality, known as location intelligence, can be used for a broad range of business applications that extend the mapping capability of Cognos 8. MapInfo provides solutions that can be directly integrated with Cognos 8. These include the ability to dynamically create geographic filters and custom areas for aggregating data for ad-hoc analysis. Examples of business applications of location intelligence are listed in the following table.

Business application	Benefits
Target marketing	Learn who your best clients are and find more like them.

Business application	Benefits
Network optimization and site location analysis	Put stores near your customers and look for gaps in geographical coverage.
Routing and work force optimization	Reduce the number of trucks you need and make your drivers more efficient.
e-government	Provide citizens with self-service opportunities.
Sales territory creation	Create balanced sales territories.
Economic development	Plan the development of your community.
Communications network planning	Avoid costly mistakes by putting cell towers in the right locations. Identify the locations of clients in your service area.

You can contact MapInfo for both data and location intelligence solutions through their Web site: www.mapinfo.com.

Repeaters

Use repeaters to repeat items when you run the report. For example, you can use repeaters to create mailing labels, including customer names and addresses.

To build a repeater, drag the **Repeater** or **Repeater Table** object from the **Toolbox** tab to the work area. Use repeaters to repeat items across a single row without a particular structure. For example, you want to create a list that contains Year, and Product line. For each year, you want all product lines to appear in a single row. To do this, create a list with Year as a column and with a repeater as a second column. Then insert Product line into the repeater. Use repeater tables to repeat items in a table structure. Drop the items you want in the repeater, and modify the properties of the repeater to obtain the results you want. For example, you can specify how many frames appear per page in a repeater table by typing values in the **Across** and **Down** properties.

Convert a List into a Repeater

You can convert a list into a repeater table to take advantage of an existing list.

Steps

1. Click any part of the list.
2. From the **Structure** menu, click **Convert List to Repeater**.

Example - Create Mailing Labels

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create mailing labels for all of the company's retailers.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Repeater Table** and click **OK**.
5. Click the repeater, click the select ancestor button in the **Properties** pane title bar, and click **Repeater Table**.
6. In the **Properties** pane, specify properties for the repeater:
   - For the **Across** property, type **2**.
   - For the **Down** property, type **5**.
   - Double-click the **Table Properties** property, select the **Fixed size** check box, and click **OK**.
7. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Table** object to the repeater.

   The **Insert Table** dialog box appears.
8. In the **Number of columns** box, type **1** and click **OK**.
9. Click the table, ensure that you see **Table Cell** in the **Properties** pane title bar, and modify the following properties:
   - Double-click **Background Image**, click **Browse**, click **logo.jpg**, and click **OK**.
   - In the **Position** box, click the align top right button.
   - In the **Tiling** box, click **Do not tile** and click **OK**.
   - Double-click **Size & Overflow**, and in the **Height** box, type **175**, and click **OK**.
10. Click the table, click the select ancestor button in the **Properties** pane title bar, and click **Table**.
11. In the **Properties** pane, specify properties for the table:
    - Double-click **Border**.
    - In the **Color** box, click **Black**.
    - In the **Style** box, click **Solid line**.
    - In the **Width** box, click **1 pt**.
    - Under **Preview**, click the all borders button and click **OK**.

- Double-click **Font**, and under the **Size** box, click **8pt**, and click **OK**.

12. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Block** object to the repeater 8 times to create 8 blocks.

13. Drag the **Text Item** object to the first block:
    - In the **Text** dialog box, type **To:** and click **OK**.
    - Click the text item.
    - Click the font button, click **Bold**, and click **OK**.

14. Click the first block, and, in the **Properties** pane, specify properties for the first block:
    - Double-click **Padding**, type **35** in the right box, click **mm** as the unit, and click **OK**.
    - Click **Horizontal Alignment** and click **Center**.

15. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)**, **Sales (query)**, and **Retailer site** and drag the seven data items to the remaining seven blocks.

16. Ctrl+click the seven blocks to select them, and in the **Properties** pane, specify properties for the seven blocks:
    - Double-click **Padding**, type **25** in the left box, click **mm** as the unit, and click **OK**.
    - Click **Horizontal Alignment** and click **Left**.

When you run the report, each page contains 10 mailing labels in two columns.

Chapter 4: Formatting a Report

Format your report to make it more readable and to reflect company standards. When you format a report in Report Studio, the formatting is stored in a layout.

You can also format a report based on conditions (p. 254).

When formatting a report, you can

- add a header or footer
- add borders
- add text
- specify the font
- add color
- reuse a style
- insert an image
- create a table of contents
- add a bookmark
- insert other objects
- align an object
- use tables to control where objects appear
- apply padding
- set margins
- reuse a layout object
- add a page
- reorder columns
- rename a row or column
- swap columns and rows
- insert page numbers
- set object properties
- set report properties
- create and modify classes

Recommendation - Laying Out a Report

When creating a report, a good layout is essential to ensure that the information in the report is presented in a clear and effective manner.

The challenge in laying out a report is in performing a mental translation of a desired layout to the layout objects available in Report Studio. Use the following steps to help you perform this translation:

- Define the page structure.

  Determine what goes into the page header, body, and footer. The page header contains information that appears at the top of each page. The page body contains information that starts on the first instance of the page. If there is too much data to fit on a single page, it continues across all instances of the page. The page footer is similar to the page header, except the information appears at the bottom of each page.

- Identify horizontal bands of information.

  Look for natural bands of information running across the page. Each of these bands typically translates into a block (p. 143).

- Identify vertical bands of information.

  In each horizontal band of information, look for bands that run up and down the page. Each of these bands typically translates into table cells (p. 147).

- Decide which data frames to use to lay out the data.

  Choose a list, crosstab, chart, repeater, or text frame.

Report Layout Guidelines

To work efficiently in Report Studio, follow these guidelines when formatting reports:

- Set properties at the highest level item possible.

  By setting properties at the highest level, you set them once instead of setting them for each child object. For example, if you want all text items in a report to appear in a specific font, set the font for the page.

  Tip: When setting properties for an object, click the select ancestor button in the **Properties** pane title bar to see the different levels available.

- Use padding and margins to create white space.

  Do not use fixed object sizing unless it is absolutely necessary. When you specify that an object has a fixed size, your layout becomes less flexible.

The Page Structure View

When you add objects to a report, you usually work in the layout. From the **View** menu, click **Page Structure** to view the report in a different way. Use the page structure view

- to view the entire contents of a report page in a tree structure

  Using a tree structure is useful for locating the objects in a page and troubleshooting problems with nested objects.

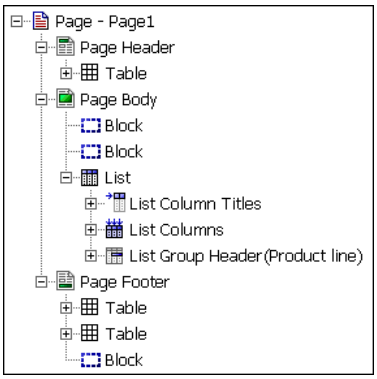
- to quickly move objects from one area of a page to another

  If you have a complex layout, it may be difficult to select, cut, and paste objects in the layout view. Objects are easier to locate in the page structure view.

- to modify object properties

  You can modify object properties in the layout or in the page structure view.

Tip: To switch back to the report layout, from the **View** menu, click **Page Design**.



Add a Header or Footer

Add a header or footer to make a report easier to read. Headers and footers are containers in which you can add objects like text, images, and report expressions such as the current date and page numbers. You can add headers and footers to pages and lists.

Pages

Add a page header or footer when you want information to appear on every page in the report, such as a title or page numbers.

Lists

You can add the following headers and footers to lists to organize data into logical sections or to identify every change in value of a column.

Type	Description
List page header	Adds a header that appears at the top of the list on every page in which list data appears.
Overall header	Adds a header that appears once at the top of the list.

Type	Description
Group or section header	Adds a header that appears for each group of a grouped column (p. 80) or each section (p. 180).
Group or section footer	Adds a footer that appears for each group of a grouped column (p. 80) or each section (p. 180).
Overall footer	Adds a footer that appears once at the bottom of the list.
List page footer	Adds a footer that appears at the bottom of the list on every page in which list data appears.

In addition, you can modify list headers and footers. You can

- split a header or footer into multiple cells that span the list (**Structure** menu, **Headers & Footers**, **Split List Row Cell**)
- merge multiple cells in a header or footer (**Structure** menu, **Headers & Footers**, **Merge List Row Cells**)
- add rows to a header or footer (**Structure** menu, **Headers & Footers**, **Insert List Row Cells Above** and **Insert List Row Cells Below**)

You can also add section headers and footers (p. 180), and you can add footers by adding a summary (p. 182).

To see the headers and footers, ensure that the visual aids are turned on (p. 52).

Steps

1. If you want to add a page header or footer, from the **Structure** menu, click **Headers & Footers**, **Page Header & Footer**, select the appropriate check boxes, and click **OK**.

   Tip: You can also click the page header & footer button on the toolbar.

2. If you want to add a list header or footer, from the **Structure** menu, click **Headers & Footers**, **List Headers & Footers**, select the appropriate check boxes, and click **OK**.

   Tip: You can also click the list headers & footers button on the toolbar.

3. If you want to add objects to a header or footer, drag the object that you want from the **Insertable Objects** pane to the appropriate location.

   To add objects to a list header or footer, you must first unlock the report. From the **Structure** menu, click **Lock Page Objects**.

   Tip: To add data items to the page header or footer, you must associate a query to the page (p. 151).

Add Borders

You can add borders to objects in a report such as a column, a header, a footer, or to the whole report.

Steps

1. Click the object to which you want to add a border.

   Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button in the title bar of the **Properties** pane.

2. In the **Properties** pane, double-click the **Border** property.

3. Select the values you want in the **Color**, **Style**, and **Width** boxes.

   Note: Specifying a width using % as the unit of measure is not supported when producing reports in PDF.

4. Apply the selected values to the top, bottom, left side, right side, or all sides of the object by clicking the appropriate button in the **Preview** section.

   Tip: To clear the border applied, click the clear borders button.

5. Click **OK**.

Tip: You can also add borders using the Report Studio toolbar.

Add Text

You can add text to a report. You can insert text in other objects, such as a block or table cell, or directly in the report page.

You can also add multilingual text to a report (p. 265).

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.

2. Drag the **Text Item** object to the report.

   The **Text** dialog box appears.

3. Type the text that you want.

   Tip: You can also paste text from another part of the report.

4. Click **OK**.

   If you click **OK** without typing any text, Report Studio inserts the string **Double click to edit text**.

You can now format the text by changing the font, color, size, and so on. Select the text and make the appropriate changes in the **Properties** pane.

Specify the Font

Specify the font in which you want text in a report to appear.

Steps

1. Click the object for which you want to specify the font.

   Tip: To specify the default font for the report, click the page.

2. Do one of the following:

   - From the toolbar, specify the font properties that you want.
   - In the **Properties** pane, double-click the **Font** property, specify the font properties that you want, and click **OK**.

     Tip: Type a list of fonts in the **Family** box if you are not sure whether a specific font is installed on a user's computer. For example, if you type **'Times New Roman', Arial, monospace**, Report Studio checks to see if Times New Roman is installed. If it is not, Report Studio checks for Arial. If Arial is not installed, the monospace font used by the computer is used.

If you clicked (**Default**) for any of the font properties, the default value for the property is used. Default values are stored in a style sheet that is used across all Cognos 8 tools. You can modify default values by modifying classes (p. 164).

Add Color

You can add background and foreground color to objects in the report.

Steps

1. Select the object to which you want to add color.

   Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button in the title bar of the **Properties** pane.

2. In the **Properties** pane, select **Background Color** or **Foreground Color**, and then click the ellipsis (...) button.

3. To apply an existing color, click the **Named Colors** tab or **Web Safe Colors** tab and choose one of the available colors.

   Use a named color to select a color from a small set of colors. Use Web safe colors to select from 216 available colors.

4. To apply a custom color, click the **Custom Color** tab and type values in the **Red**, **Green**, and **Blue** boxes.

   The values must be hexadecimal.

5. Click **OK**.

Copy and Reuse Formatting

You can quickly copy the formatting of items in your report, such as fonts, colors, borders, and number formats and apply that formatting to other items.

Steps

1. Click an item that has the formatting that you want to copy.
2. Do one of the following:
   - To copy all the formatting applied to the item, click the pick up style button, click the item that you want to format, and then click the apply style button.
   - To copy only one of the formatting styles, click the down arrow to the right of the pick up style button and click the style that you want to copy. Then, click the item that you want to format and click the apply style button.
3. If you want to make changes to a style that you copied, click the down arrow to the right of the pick up style button and click **Edit Dropper Style**. In the **Style** dialog box, specify basic and advanced style characteristics, and then click **OK**.

Insert an Image

You can insert an image in a report. You can insert images in other objects, such as a block or table cell, directly in the report page, or as the background image of another object.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag the **Image** object to the report and then double-click it.
3. In the **Image URL** dialog box, type the URL of the image you want to insert or click the **Browse** button to go to the location containing the image.

   To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server. For more information, see "Troubleshooting" (p. 317).
4. Click **OK**.

Insert a Background Image

You can insert a background image for objects in a report. For example, use a background image to add a watermark to a page.

Steps

1. Select the object for which you want to add a background image.

2. In the **Properties** pane, double-click the **Background Image** property.
3. In the **Image URL** box, type the URL of the image you want to insert or click the **Browse** button to go to the location containing the image.

   The image appears in the **Preview** box.
4. In the **Position** box, choose how you want to align the image in the object.
5. In the **Tiling** box, click the tiling option you want.
6. Click **OK**.

Create a Table of Contents

You can create a table of contents that appears in the rendered output of your report. A table of contents is useful for reports that include sectioned items, grouped items, or multiple pages in the layout. The report output indicates page numbers and allows for easy navigation.

You can include multiple tables of contents in your report, which are useful if your report contains grouped lists. You can also add section numbers in front of entries in the table of contents by modifying the heading level property.

Table of contents entries are logical markers placed anywhere in a report. For example, you can place entries at the top of a page or in a list group header to mark each grouped data value. Although table of contents entries are visible in Report Studio, they cannot be seen when a report is run.

A table of contents works only for reports produced in PDF and non-interactive HTML format. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a table of contents works only if the target exists in the page currently being viewed.

If you want to quickly move from one part of a report to another without using page numbers, you can add bookmarks (p. 141).

You must first create a table of contents before adding entries in the report. All entries must be inserted after the table of contents in the report layout.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag the **Table of Contents** object to the location you want, which can be anywhere in the report.

   A table of contents placeholder appears.
3. Drag the **Table of Contents Entry** object to the location where you want to locate your first table of contents marker.

   Tip: You can also click **Structure**, and then **Insert Table of Contents Entry**.

   The new entry appears in the table of contents.
4. Double-click the **Double click to edit text** box of the new marker.

5. In the **Text** box, type the text to appear in the table of contents, and then click **OK**.
6. To edit the heading level of a table of contents entry, click the entry, and in the **Properties** pane, click the **Heading Level** property and select the level.

   The heading level is used to insert section numbers in front of entries in the table of contents using layout calculations.

When you finish creating the table of contents, click the run report button . By clicking the arrow to the right of the run report button, you can specify whether you want to run the report as HTML or PDF.

Add a Bookmark

Add a bookmark so that users can quickly move from one part of a report to another. For example, a list report contains many rows of data. You use bookmarks so that users can move to specific rows.

You can also drill through to another report (p. 273).

If you want to use page numbers to move from one part of a report to another, you can also create a table of contents (p. 140).

Bookmarks work for reports produced in HTML format or PDF. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a bookmark works only if the target exists in the page currently being viewed.

Tip: You can reduce the number of HTML pages generated when a report is run interactively by specifying a value for the **Rows Per Page** property for a data container in the report.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag **Bookmark** to the location you want, which can be anywhere in the report.
3. Click the bookmark, and in the **Properties** pane, click the **Source Type** property and then click a source type.

Source type	Description
Text	Creates a static bookmark using a text value. For example, a list has sections, and you want users to jump from each section to the top of the report.
Report Expression	Creates a dynamic bookmark whose values are derived from an expression that you define.

Source type	Description
Data Item Value	Creates a dynamic bookmark that has data as possible values. This is useful for creating a context-based bookmark. For example a list has sections, and you want users to jump from the top of the report to a specific section. **Note:** This source type appears only if the bookmark is inserted next to a data item.
Data Item Label	Creates a bookmark that has the label of a data item as its value. Use this source type to jump to the first occurrence of a data item label. For example, a list is divided into sections using Product line. You want users to jump to the first product line section that appears in the list rather than to a specific section, such as Camping Equipment. **Note:** This source type appears only if the bookmark is inserted next to a data item.
Member Caption	In a crosstab, creates a dynamic bookmark that uses member captions as possible values.
Cell Value	In a crosstab, creates a dynamic bookmark that uses cell values as possible values.

4. Click the property that appears under the source type you chose and specify the bookmark value.

   For example, if the source type is **Data Item Value**, click the **Data Item Value** property and click the data item you want to use.

5. Right-click the object that you want to use to jump to the bookmark and click **Drill-Through Definitions**.

   For example, right-click a data item, image, chart, or text item.

   Tip: The object can exist in a different report, so you can jump from one report to another.

6. Click the new drill through button .

7. Click the **Bookmark** tab.

8. Click **Source type** and click one of the source types described in step 4.

   Click the source type you want to use to produce the value needed to jump to the bookmark. For example, click **Data Item Values** if you want the value to come from a data item, such as Product line.

9. If you clicked one of the following source types, specify the value to use to jump to the bookmark.
   - For **Text**, click the ellipsis (...) button next to **Text**, and type a text value.
   - For **Data Item Value** or **Data Item Label**, click the **Data item** list, and choose a data item.
   - For **Report Expression**, click the ellipsis (...) button next to **Report expression**, and define the expression.
10. Click **OK**.

Insert Other Objects

In addition to text and images, the **Toolbox** tab in the **Insertable Objects** pane contains other objects that you can add to the report layout.

Object	Description
Block	Adds an empty block, a container in which you can insert other objects. This is useful for controlling where objects appear. **Tip:** You can use blocks to add space between objects. However, empty blocks are not rendered. You must insert an object or specify the height and width.
Table	Adds a table, a container in which you can insert other objects. This is useful for controlling where objects appear.
Field Set	Adds an empty block that has a caption. This is similar to the **Block** object, but with a caption.
Calculated Member	Adds a calculated member (p. 190).
Intersection (Tuple)	Adds an intersection (tuple) (p. 191).
Query Calculation	Adds a calculated column (p. 189).
Layout Calculation	Adds a calculation in the layout that contains run-time information, such as current date, current time, and user name.
Crosstab Space	Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. Blank cells appear for the edge when the report is run. Insert this object when a crosstab edge does not produce useful data and you want blanks to appear in the cells instead.

Object	Description
Crosstab Space (with fact cells)	Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. The contents of the fact cells for the edge are rendered when a measure is added or the default measure (p. 198) is specified. • If the crosstab space is nested, the scope of the fact cells is the scope of the item that is at the level before the space. • If the crosstab space is not nested and there are no items nested below it, the scope of the fact cells is the default measure.
Singleton	Inserts a single data item (p. 196).
Conditional Blocks	Adds an empty block that you can use for conditional formatting (p. 254).
HTML Item	Adds a container in which you can insert HTML code. HTML items can be anything that your browser will execute, including links, images, multimedia, tooltips, or JavaScript. HTML items appear only when you run the report in HTML format. **Note:** You cannot include <form> tags in HTML items. When you upgrade to the next version of Cognos 8, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features such as JavaScript that refers to Cognos HTML objects.
Rich Text Item	Inserts an object that is used to render HTML in the layout. This object is similar to the **HTML Item**, except that rich text items also render in PDF output. Using rich text items is useful when you want to add annotations defined in a data source to a report. **Note:** Rich text items support only a restricted set of well-formed XHTML.
Hyperlink	Adds a hyperlink so that users can jump to another place, such as a Web site.
Hyperlink Button	Adds a hyperlink in the form of a button.

Object	Description
As of Time Expression	Adds a As of Time expression so that you can show data for a specific time period (p. 197).
Page Number	Inserts page numbers that you can customize (p. 157).
Row Number	Numbers each row of data returned when the report is run. **Note:** You can add row numbers only to lists and repeaters.
Layout Component Reference	Adds a reference to another object. Useful when you want to reuse an object.
Metric Studio Diagram	Adds a Metric Studio history chart as an image. For information about adding a Metric Studio diagram to a report, see the Metric Studio *User Guide*.
Table of Contents	Creates a table of contents that is rendered in the output of a report (p. 140).
Table of Contents Entry	Adds a table of contents marker (p. 140).
Bookmark	Inserts a bookmark so that users can move from one part of a report to another (p. 141).

For information about each report type, see "Types of Reports" (p. 79). For information about each prompt control and prompt buttons, see "Adding Prompts" (p. 199). The **Toolbox** tab contains other objects when you work in other areas such as Query Explorer (p. 213) and Condition Explorer (p. 259). You can also add your own objects to the toolbox, such as a company logo that you use often. For more information, see the *Administration and Security Guide*.

Before you can add a hyperlink, HTML item, or hyperlink button, you must have the **HTML Items in Report** capability. For more information, see the *Administration and Security Guide*.

Step

- To add an object, drag or double-click it from the **Toolbox** tab .

Elements Supported in Rich Text Items

Rich text items support all XHTML character entities, such as   (non-breaking space), and the following elements:

- div
- span

- ul
- ol
- li

Each element only supports the style attribute, which must contain a valid CSS style. In addition, ul and ol elements support list-style attributes. Specifically, the ol element supports decimal, and the ul element supports circle, disc, and square, as well as list-style-image.

For example, the following code produces an unordered list entitled List: with three items. Each list item is in a different color, and the list-style attribute used is circle.

List: <ul style="list-style-type:circle"> <li style="color:green">Item A <li style="color: red">Item B <li style="color:blue">Item C

Example - Add a Multimedia File to a Report

You are a report author at the Great Outdoors Company, which sells sporting equipment. You want to insert a Windows Media Audio/Video file named GO.wmv in a template that serves as a cover page for all reports.

You must have Windows Media Player installed on your computer.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the top right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template** and in the **New** dialog box, click **Blank**.
4. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **HTML Item** object to the report.
5. In the report, select the **HTML Item** you just added.
6. In the **Properties** pane, double-click the **HTML** property.
7. In the **HTML** dialog box, type the following:

```
<OBJECT classid="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6">
    <PARAM NAME="URL" VALUE="/c8/webcontent/samples/images/GO.wmv"/>
</OBJECT>
```

8. Click **OK**.

When you run the report, the multimedia file plays in Windows Media Player.

Align an Object

You can specify a horizontal and vertical alignment for an object in a report to determine where they appear.

Tables can also be used to determine where objects appear in a report.

Note: The **Justify** horizontal alignment works with HTML output but does not apply to PDF output.

Steps

1. Select the object that you want to align.
2. From the toolbar, click one of the available horizontal or vertical alignment buttons.

Use Tables to Control Where Objects Appear

You can use tables in your report to control where objects appear. Tables can be inserted anywhere in a report, such as a header, a footer, or the page body. After you create a table, insert the objects you want in the cells.

You can also apply a predefined table style to tables.

The alignment buttons can also be used to determine where objects appear in a report.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag **Table** to the report.

   The **Insert Table** dialog box appears.
3. In the **Number of columns** and **Number of rows** boxes, type the number of columns and rows you want the table to have.
4. If you want to have the table span the width of the report page, select the **Maximize width** check box.
5. If you want to add borders to the table, select the **Show borders** check box.
6. Click **OK**.

Apply a Table Style

Apply a table style to quickly format tables. You can also apply a table style to lists, crosstabs, and repeater tables.

Steps

1. Click the table, list, or crosstab to which you want to apply a table style.

2. From the **Table** menu, click **Apply Table Style**.
3. In the **Table styles** box, click the table style that you want.

   Tip: Some styles are unique to tables, lists, or crosstabs.
4. If you are applying a table style to a table or list, in the **Apply special styles to** section, select or clear the various column and row check boxes based on how you want to treat the first and last columns and rows.

   Some check boxes may not be available for particular table styles, or to particular columns or rows.
5. If you are applying a table style to a list or crosstab, select the **Set this style as the default for this report** check box to set the style as the default for all lists and crosstabs.

   For a list, you may need to clear the **First column** and **Last column** check boxes in the **Apply special styles to** section before you can select this check box. In addition, some tables styles cannot be set as the default.
6. Click **OK**.

Apply Padding

Apply padding to an object to add white space between the object and its margin or, if there is a border, between the object and its border.

Steps

1. Select the object to which you want to apply padding.
2. In the **Properties** pane, double-click the **Padding** property.
3. Specify top, bottom, left, and right padding by typing values in the corresponding boxes and choosing the unit of measure you want.

   Note: Specifying padding using % as the unit of measure is not supported when producing reports in PDF.
4. Click **OK**.

Tip: You can quickly apply left or right padding by either pressing Tab and Shift+Tab or by clicking the increase indent and the decrease indent buttons in the toolbar. When using the toolbar buttons, you can indent an object by up to nine times the indentation length. You can specify the indentation length to use by clicking the arrow beside either button and clicking **Set Default Indent Length**.

Set Margins

Set the margins for objects in a report to add white space around them.

For Date, Time, Row Number, and Page Number objects, you can only set the left and right margins. If you want to set the top or bottom margins for these objects, place them in a table or a block. Then set the margin or padding properties on the table or block object.

Steps

1. Select the object for which you want to set margins.
2. In the **Properties** pane, double-click the **Margin** property.
3. Specify the top, bottom, left, and right margins by typing values in the corresponding boxes and choosing the unit of measure you want.

   Note: Specifying margins using % as the unit of measure is not supported when producing reports in PDF.
4. Click **OK**.

Reuse a Layout Object

You can save time by reusing layout objects that you add to a report instead of re-creating them. For example, you have a multiple-page report and you want to show the company logo in the page header of each page. Insert the logo once and reuse it on all other pages.

Steps

1. Click the object that you want to reuse.

   Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button in the title bar of the **Properties** pane.
2. In the **Properties** pane, in the **Name** property, type a value beginning with a letter to uniquely identify the object and press the Enter key.

   Report Studio may have already specified a name for the object.
3. To reuse the object in another report, open that report.
4. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Layout Component Reference** object to the location where you want to reuse the object.
5. In the **Component Location** box, do the following:
   - Click **This report** to reference an object in the current report.
   - Click **Another report** to reference an object in another report, click the ellipsis (...) button, and open the report you want.
6. In the **Available components to reference** box, click the object you want and click **OK**.
7. If the referenced object is in another report, click the **Layout Component Reference** object and in the **Properties** pane, click the **Embed** property and specify how to store the referenced object in the report:

- Click **Copy** to store a copy of the object.

  The copy is not automatically updated if the source object is modified.

- Click **Reference** to store a reference, or pointer, of the object.

  The reference of the object is automatically updated if the source object is modified. For example, if you open or run the report, you see the modified object.

A copy or reference of the object appears where you placed the **Layout Component Reference** object. If a source object is changed, you can update reused objects.

Tip: You can also create a new report or template and add all the objects you want to share. All your shared objects then reside in a single location, like a library.

Change a Reused Object

If you reuse an object that contains other objects, you can change the child objects to something different. For example, you have a block object containing a text item in the page header, and you decide to reuse the block in the page footer. However, you want the text item in the page footer block to show different text from that in the page header.

Steps

1. In the parent object that you want to reuse, click the child object you want to change.
2. In the **Properties** pane, in the **Name** property, type a value beginning with a letter to uniquely identify the object.

   Report Studio may have already specified a name for the object.
3. Select the copy of the parent object you created with the **Layout Component Reference** object.
4. In the **Properties** pane, double-click the **Component Overrides** property.
5. In the **Component Overrides** dialog box, select the child object you want to change and click **OK**.

   The child object in the copy of the parent object is replaced by the following text:

   Drag & drop object to override component child.
6. Drag the object that you want to replace the child object.

   You can replace the child object with any other object, not just an object of the same type. For example, if the child object is a text item, you can replace it with an image.

Update Reused Objects

If a report contains objects referenced in another report, you can quickly update the referenced objects if the source objects have changed. Shared objects are stored in the layout component cache.

Steps

1. Open the report that you want to update.
2. From the **Tools** menu, click **Layout Component Cache**.

   The **Layout Component Cache** dialog box appears, showing all reports that contain referenced objects and the source reports where the objects exist.
3. To view which components are reused, click a source report.

   The components that are reused appear in the **Components used** pane.
4. Click the **Reload Components** button to refresh all referenced objects.

   Although referenced objects are automatically refreshed when you open or run a report, clicking this button updates components that were changed while the report is open.
5. Click the **Update All Component Copies** button to refresh all copied objects.
6. Click **Close**.

Add a Page

Reports may contain pages with a variety of content. Pages may be added to create a multiple-page report. You may want to do this for a variety of reasons. For example, you are creating a sales report and you want to include general information about the company. You decide to create an introduction page to show this information. When you run the report, page 1 of the report is the introduction page. The following pages contain data.

You can also create page sets to control the rendering of pages in a report.

Steps

1. Pause the pointer over the page explorer button and click **Report Pages**.

   Tip: Click **Prompt Pages** to add a prompt page (p. 199).
2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Page** to the work area.
3. To change the order of pages, click a page and drag it above or below the other existing pages.
4. To associate a query to the page, in the **Properties** pane, click the **Query** property and click the query you want.

   Associate the page to a query to add data-related objects directly to the page header, body, or footer, such as data items and filters.
5. Double-click the page to open it in page design view (p. 134).

6. Add the objects you want to the page.

Create Page Sets

Create page sets to associate pages with a query structure to force page breaks. For example, a query contains the data item Product line, which is grouped. Creating a page set that is associated with this query adds page breaks for each product line.

When working with dimensional data, you can use the **Page layers** area to show values on a separate page for each member (p. 181).

Steps

1. Pause the pointer over the page explorer button and click **Report Pages**.
2. Create the pages you want to appear in the report.
3. In the **Insertable Objects** pane, drag the **Page Set** object to the **Report Pages** pane.
4. In the **Properties** pane, click the **Query** property and click the query you want to associate to the page set.
5. Organize the pages in the report by dragging report pages to the page set you want.
6. Insert the page containing details in the **Detail Pages** folder.

   Detail pages are the pages that repeat based on the items you group by in the following step.

   Tip: You can insert multiple detail pages into the same page set, and link them using a master detail relationship (p. 282).
7. Define the grouping structure for the page set:
   - Click the page set.
   - In the **Properties** pane, double-click the **Grouping & Sorting** property.
   - In the **Data Items** pane, drag the data item you want to group by to the **Groups** folder in the **Groups** pane.
   - To sort the data (p. 177) within each group, in the **Data Items** pane, drag the data item you want to sort by to the **Detail Sort List** folder, and then click the sort order button to specify the sort order.
   - Click **OK**.
8. Repeat steps 2 to 6 to create other page sets.

   Tip: You can nest page sets, and join them by defining a master detail relationship. Create nested page sets to have pages occur within other pages. For example, you want pages containing product type information occur within pages containing product line information.

Note: Grouping an item for a page set is not the same as grouping a column in the layout (p. 80). Grouping a column in the layout visually shows groups in a report. Grouping an item for a page

set groups the item in the query. If you want to use an item that is already grouped in the layout, you must still perform step 7.

Join Nested Page Sets

If you have nested page sets in your report, define a master detail relationship between them to see data in the nested page set that is related to the data in the parent page set. For example, you have a page set that shows pages of product line information. The page set contains a nested page set that shows pages of product type information. For each product line page, you want to see the related product type pages, as shown below:

Product line_1

- Product type_1
- Product type_2
- Product type_3

Product line_2

- Product type_4
- Product type_5
- Product type_6

Steps

1. Pause the pointer over the page explorer button and click **Report Pages**.
2. In the **Report Pages** pane, click the nested page set.
3. In the **Properties** pane, double-click the **Master Detail Relationships** property.
4. Click the **New Link** button.
5. In the **Master Query** box, click the data item that provides the primary information.
6. Link the master data item to the details by doing one of the following:
   - To link to another data item in the detail query, in the **Detail Query** box, click the data item that provides the detailed information.
   - To link to a parameter, in the **Parameters** box, click the parameter that provides the detailed information.
7. Repeat steps 5 to 7 to create other links.

   Tip: To delete a link, click it and press the Delete key.
8. Click **OK**.

For more information about master detail relationships, see "Create a Master Detail Relationship" (p. 282).

Example - Preparing a Product List Report

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a multiple-page report showing all products sold by the company. You are asked to create title and end pages, and to have each product line appear on a new page, preceded by a header page and followed by a footer page.

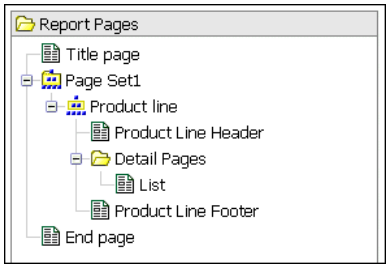
Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , add the following data items to the list by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**

   Tip: You can find these data items by expanding, **Sales and Marketing (query)**, and **Sales (query)** . These items are in the **Product** folder.
6. Expand **Sales fact** and add **Production cost** by double-clicking it.
7. Group the **Product line** and **Product type** columns.
8. Pause the pointer over the page explorer button and click **Report Pages**.
9. In the **Insertable Objects** pane, drag **Page Set** to the work area and associate it to the query **Query1**.
10. In the **Insertable Objects** pane, drag **Page** to the work area four times.
11. For each page, click the page, and in the **Properties** pane, double-click the **Name** property and name it as follows:
    - **Title page**
    - **End page**
    - **Product Line Header**
    - **Product Line Footer**
12. For each page just created, add the objects you want.

    For example, add a text item to each page to uniquely identify it.

13. Rename the **Page1** page, which contains the list, to **List**.
14. Click the page set, and, in the **Properties** pane, double-click the **Grouping & Sorting** property.
15. In the **Data Items** box, drag **Product line** to the **Groups** folder in the **Groups** box and click **OK**.
16. Organize the report pages into the following hierarchy by dragging them to the appropriate location.



When you run the report, the following pages appear:

- Title page
- Product line header page
- A page for Product line_1
- Product line footer page
- Product line header
- A page for Product line_2
- Product line footer
- ...
- End page

Reorder Rows and Columns

Change the order of rows and columns in a report to rearrange information.

Steps

1. Click the column you want to reorder in the report.

   You can click the column heading or a column.
2. Drag the column to a new location.

   A black bar indicates where you can drop the column.

Rename a Row or Column

Change the column title to provide a more meaningful name. By default, when you run a report, the column title is taken from one of the following:

- if defined, the **Label** property of the data item
- if defined, the label of the data item in the model
- the **Name** property of the data item

Consequently, the column title you see in the layout may not be the column title you see when you run the report. For example, if you modify the **Name** property of the data item, the column title in the layout will change to the new name. However, when you run the report, the column title that appears will be, if defined, the data item label in the report or the data item label in the model. If neither of these are defined, only then will the modified name be used as the column title.

Steps

1. Click the column heading you want to change.
2. In the **Properties** pane, click the **Source Type** property and click the source type you want to use for the column title:
   - To use a static text item, click **Text**, and then type the name you want in the **Text** property that appears.
   - To use a data item value, click **Data Item Value**, and then click the data item you want to use in the **Data Item Value** property that appears.

     Using a data item value is useful when you have a master detail relationship. For example, you create a list with sections using a master detail relationship (p. 180), and you want the section name to appear in the column title of one of the columns. Insert the data item used as section headings into the column title, and specify the source type as **Data Item Value**. When you run the report, for each section, the column title will contain the section heading combined with the data item label or name.
   - To use a label that is different from the label in the model, click **Data Item Label**, click the data item you want to use in the **Data Item Label** property that appears, and type the name you want in the **Label** property.

     Tip: The **Label** property is under the **Data Item** group.
   - To use a report expression, click **Report Expression**, double-click the **Report Expression** property that appears, type the expression you want, and click **OK**.
3. Press Enter.

Swap Columns and Rows

Swap columns and rows to look at information from a different perspective. This may help you discover high and low points in the data that you hadn't previously noted.

You can only swap columns and rows in a crosstab or chart.

Step

- From the toolbar, click the swap rows and columns button .

In the report, the rows become the columns and the columns become the rows.

Insert Page Numbers

You can insert page numbers in a report and specify the number style to use. You can select a predefined page numbering scheme or create a custom scheme. You can easily insert page numbers using the **Page Number** object.

You can also manually create an expression to insert page numbers by inserting a layout calculation and using the different page report functions (p. 189) in the expression editor.

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Page Number** to the location where you want page numbers to appear.

   Tip: When you create a new report using one of the existing report templates, **Page Number** is already inserted in the page footer.

2. Right-click the page number symbol and click **Edit Number Style**.

3. Choose the style that you want to use.

   The first three choices apply only to vertical page numbers. The remaining choices specify how vertical and horizontal page values appear.

4. If you want to customize the choice that you made in the previous step, click the edit button , make the changes that you want, and click **OK**.

   A custom number style is created. If you later choose a different number style, the custom style is removed from the list.

   Tip: In the **Custom Number Style** dialog box, when you pause the pointer over a box, a tooltip describes how that box affects page numbers. For example, the **Separator Text** box contains the text, such as a hyphen, that separates page values for both vertical and horizontal pages.

Setting Object Properties

For each object you insert in a report, you can define its properties by specifying values in the **Properties** pane. Some properties, such as borders, color, and padding, are discussed elsewhere. Other properties include

- pagination
- spacing and breaking
- text flow and justification
- size and overflow
- floating
- table properties

Some properties apply only to some objects. For more information about objects and properties, see "Report Studio Object and Property Reference" (p. 523).

Many object properties are Cascading Style Sheets (CSS) attributes. For information about a particular property that is a CSS attribute, see the CSS documentation on the World Wide Web Consortium (W3C) Web site (http://www.w3.org).

Note: Some properties do not apply to all types of report output. For example, the **Direction** text flow property works with HTML output but does not apply to PDF output.

Control Page Breaks and Page Numbering

You can control page breaks and page numbering in a list, crosstab, table, or report page by choosing any of these options.

Option	Description
Keep with header	Keeps all headers on the same page with the number of detail rows specified.
Keep with footer	Keeps all footers on the same page with the number of detail rows specified.
Reset page count	Resets the page count after a page break to the value specified.
Reset page number	Resets the page number after a page break to the value specified.
Repeat every page	If the report renders multiple pages, this object is repeated on every page.
Allow contents to break across pages	Allows contents to break across pages.

Option	Description
Allow horizontal pagination	In PDF output, allows the columns of a list or crosstab to break across horizontal pages if they do not fit on a single page. **Tip:** In lists, you can select the **Repeat every page** option for list columns that you want to show on every horizontal page. If the **Allow horizontal pagination** option is not selected, the size of the list or crosstab is scaled down when necessary so that it fits on a single page. If your report includes nested data frames such as a list within a list, horizontal pagination is supported on either the parent or child frame, but not both. If horizontal pagination is enabled on both the parent and child frame, it will be ignored on the child frame when the report runs. We recommend that you do not enable horizontal pagination on both the parent and child frames. You can also specify page number options that use compound numbering schemes. For example, you can use the numbering scheme 1-1, 1-2, 2-1, 2-2, and so on. For more information, see "Insert Page Numbers" (p. 157).
Enable horizontal page numbering	Increments page numbers of horizontal pages separately from the main page numbers when you select a page numbering style (p. 157) that includes horizontal pages. For example, if a page has three page breaks horizontally and you selected the page number style 1a, the horizontal pages are numbered 1a, 1b, and 1c. If you did not select a numbering style that includes horizontal pages, the horizontal pages are all numbered 1 for the first vertical page, 2 for the second vertical page, and so on. If this option is not selected and there are horizontal pages, all pages are numbered sequentially. For example, if a report has two vertical pages and three horizontal pages, the PDF pages are numbered from 1 to 6. Pages 1 to 3 are the three horizontal pages for the first vertical page and pages 4 to 6 are the three horizontal pages for the second vertical page.
Repeat table rows on page break	In tables, if a row breaks across pages, repeats the rows that were previously rendered on each page. By default, table rows are repeated. **Note:** This option applies to saved reports only. In interactive HTML reports, table rows are always repeated even if this option is not selected.

You can also specify the style to use for page numbers (p. 157).

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Pagination** property.
3. Specify how you want pages to break and be numbered by specifying the options that you want.

Specify Text Properties

You can specify text properties by choosing any of these options.

Option	Description
Line Height	Sets the distance between lines of text in an object.
Letter Spacing	Sets the amount of additional space between letters in an object.
Text Indent	Sets the indentation of the first line of text in an object. **Note:** This property works with HTML output but does not apply to PDF output.
Word Break	Sets line-breaking behavior within words.
Break Words when Necessary	Sets whether to break words when the content exceeds the boundaries of an object.
Enforce Stricter Line-Breaking Rules for Japanese Text	Sets line-breaking rules for Japanese text.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Spacing & Breaking** property.
3. Specify the text properties that you want.

Specify Text Flow

You can specify text flow properties by choosing any of these options.

Option	Description
Direction	Sets the reading order of an object, such as right to left.
Writing mode	Sets the direction and flow of content in an object.
Bi-directional	Sets the level of embedding in an object.
Type	Sets the type of alignment used to justify text in an object.
Kashida space	Sets the ratio of kashida expansion to white space expansion when justifying lines of text in the object. This property is used in Arabic writing systems.

Note: All the text flow properties work with HTML output but do not apply to PDF output.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Text Flow & Justification** property.
3. Choose how you want text to flow by specifying the options that you want.

Specify the Height and Width

You can specify the height and width of objects using various units of measurement. In addition, if the object is a field set, text box prompt, prompt button, hyperlink button, block, or a conditional block, you can specify how to handle content overflow. Specify the height and width by choosing any of these options.

Option	Description
Height	Sets the height of the object.
Width	Sets the width of the object.
Content is not clipped	If the contents of the block exceed the height or width of the block, the block automatically resizes to fit the contents.
Content is clipped	If the contents of the block exceed the height or width of the block, the content is clipped. **Note:** The clipped content still exists. It is just not visible in the block.

Option	Description
Use scrollbars only when necessary	If the contents of the block exceed the height or width of the block, scrollbars are added to the block.
Always use scrollbars	Scrollbars are added to the block.

Do not use percentages to resize charts and maps that contain interactive elements that are activated when you pause the pointer over them, such as tooltips or drill-through links, because the browser is unable to realign the hard-coded hot spots after an image is resized.

When you use a percentage to specify the size of an object, the percentage is relative to the object's parent. In some cases, setting the size of an object using percentages will not give you the results that you want unless you also specify the size of the parent container.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Size & Overflow** property.
3. Specify the options that you want.

Control How Objects Flow Around Other Objects

You can control how objects flow around other objects by choosing any of the following options.

Option	Description
Float	Sets how other objects flow around the object.
Allow floating objects on both sides	Allows other objects to flow on both sides. The Float property must be set.
Move below any floating object on the left side	If there are other objects to the left of the object, the object moves below those objects. The Float property must be set.
Move below any floating object on the right side	If there are other objects to the right of the object, the object moves under those objects. The Float property must be set.

Option	Description
Move below any floating object	Moves the object under any other object in which the Float property was set.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Floating** property.
3. Specify how you want other objects to flow around the object by specifying the options that you want.

Specify Borders and Cell Size in Tables

You can specify the borders and cell size in lists, crosstabs, and tables by choosing any of these options.

Option	Description
Collapse borders	Collapses adjacent cell borders into a single border.
Cell spacing	If borders are not collapsed, sets the amount of space between cell borders.
Show empty cell borders	If borders are not collapsed, shows borders of cells that are empty. **Note:** This property works with HTML output but does not apply to PDF output.
Fixed size	Makes all cells the same fixed, size.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Table Properties** property.
3. Specify the borders and cell size by specifying the options that you want.

Specify Report Properties

You can change the default report property settings.

Steps

1. From the **File** menu, click **Report Properties**.
2. To specify the classes that Report Studio uses to format objects, click **Report styles** and choose one of the available options.

   For more information, see "Create and Modify Classes" (p. 164).
3. For reports with multiple data containers, to render the default number of rows of each data container on each HTML page, you must set the **Page break by data container for interactive HTML** option to **Yes**. The default value is **No**.

   For more information, see "Controlling the Rows Per Page for Multiple Containers in HTML and PDF" (p. 70).
4. To automatically create extended data items every time a data item is inserted, click **Always create extended data items**.

   If this property is not selected, Report Studio creates expression-based data items. Expression-based data items allow you to view or edit the expression by double-clicking the **Expression** property for the item.

Create and Modify Classes

Create your own classes or modify existing classes in a report to format objects across a report according to your particular needs. In Report Studio, objects in reports are assigned a Cascading Style Sheet (CSS) class that provides a default style for the object. For example, when you create a new report, the report title has the class property **Report title text** assigned to it. In addition, objects inherit the classes set on their parent objects.

You can use classes to highlight data using conditional styles (p. 254).

Classes you create or modify can be applied only to the current report. To create or modify classes for all reports, you must modify a layout style sheet. In addition, some classes can be used to format Query Studio reports.

Steps

1. From the **File** menu, click **Report Properties**.
2. Click **Report styles** and select one of the following options:
   - To work with classes in the default style sheet, click **Default styles**.
   - To work with classes that were used in Cognos ReportNet, click **Use 1.x report styles**.

Use 1.x report styles when you are working with reports created in ReportNet and you want to preserve their original appearance.

- To work with classes that have minimal styling defined, click **Simplified styles**.

  This option is useful when creating financial reports.

3. Pause the pointer over the page explorer button and click **Classes**.
4. To create a new class, in the **Insertable Objects** pane, drag **Class** to the **Local Classes** pane.
5. To modify an existing class, in the **Local Classes** or **Global Class Extensions** pane, click the class that you want.

   Modify a global class to apply a change to all objects that use that class. For example, if you modified the style **List column title cell**, all column titles in lists will reflect your modifications.

   Tip: Ctrl+click classes to make the same change to more than one class.
6. In the **Properties** pane, modify the properties to specify your desired formatting.

   Tip: Look at the **Preview** pane to preview your changes for different report objects, such as blocks, table cells, and text items.

   If you modify a global class, a pencil symbol appears beside the global class icon to indicate that the class was modified.
7. Apply the class to the objects you want:
   - Pause the pointer over the page explorer button, and click a report page.
   - Click an object to which you want to apply a class.
   - In the **Properties** pane, double-click the **Class** property.
   - Click the classes that you want to apply from the **Local classes** and **Global classes** panes and then click the right arrow button .
   - If you applied more than one class, in the **Selected classes** pane, specify the order in which the classes are applied by clicking each class and clicking the up or down arrow button.

     Classes in the **Selected classes** pane are applied from top to bottom. The style properties from all classes are merged together when they are applied. However, if the classes have style properties in common, the properties from the last class applied override those from previous classes.
   - Click **OK**.

Modify Classes to Format Query Studio Reports

Some global classes are specific to Query Studio or can be applied to Query Studio reports. You can modify the following classes to format Query Studio reports.

Class	Description
List column body cell	Style used to format list data.
List column title cell	Style used to format list column headings.

Steps

1. Create a Query Studio template (p. 75).
2. Modify the global classes that you want.
3. Save the template.

To format a Query Studio report using the modified classes, the template must be applied to the report. For more information about applying a template to a Query Studio report, see the Query Studio *User Guide*.

Modifying the Default Layout Style Sheet

In addition to creating or modifying classes in a report, you can create and modify classes that will apply to all reports. Default styles are stored in a style sheet named GlobalReportStyles.css. For information about modifying the style sheet, see the *Administration and Security Guide*.

Chapter 5: Working with Data

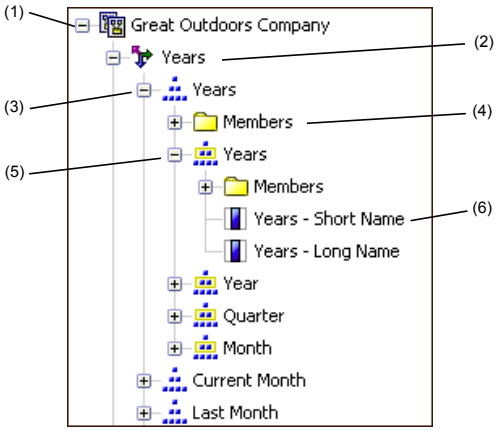
Make a report easier to read and easier to understand by specifying what data appears in the report and how the data looks. In Report Studio, you can

- work with dimensional data
- filter data
- sort data
- create sections
- summarize data
- create a calculation
- format data
- insert a single data item
- specify the list of properties for a layout object
- show data for a specific time period
- specify the default measure
- specify what appears for a data container when no data is available
- define a prompt
- work with queries

Working with Dimensional Data

If you are working with a dimensional data source, data items are organized hierarchically. Dimensional data sources include OLAP and dimensionally modeled relational (DMR) data sources.

The **Source** tab in the **Insertable Objects** pane shows a metadata-based view of the data.



Note: The names of levels and members in a dimension come from the model. It is the responsibility of the modeler to provide meaningful names.

1. Package

   Packages are subsets of a model, containing items that you can insert in a report.

2. Dimension

   Dimensions are broad groupings of descriptive data about a major aspect of a business, such as products, dates, or markets.

3. Level hierarchy

   Level hierarchies are more specific groupings within a dimension. For example, for the **Years** dimension, data can be organized into smaller groups, such as **Years**, **Current Month**, and **Last Month**.

4. Members folder

   Members folders contain the available members for a hierarchy or level. For example, the **Members** folder for the **Years** level hierarchy contains everything found in the **Year**, **Quarter**, and **Month** levels.

   Note: To see members folders, you must select the **Show members folder** check box in the **Tools** menu (p. 57).

5. Level

   Levels are positions within the dimensional hierarchy that contain information at the same order of detail and have attributes in common. Multiple levels can exist within a level hierarchy, beginning with a root level. For example, the **Years** level hierarchy has the following related levels.

Level	Level name	Description
Root	**Years**	The root level.
First	**Year**	Years in the **Years** root level. For example, 2004, 2003, and 2002.
Second	**Quarter**	Quarters for each year in the **Year** level. For example, 2004 Q1, 2004 Q2, and 2004 Q3.
Third	**Month**	Months for each quarter in the **Quarter** level. For example, Jan., Feb., and Mar.

   Tip: The **Measures** dimension contains the measures available in the data source.

6. Member property

Member properties are attributes that each member possesses. For example, gender could be a property for all employee members. For more information about member properties, see "Insert a Member Property" (p. 170).

Insert Members from the Tree

By default, when you insert members from the source tree into your report, single members are inserted without any details or children. You can change how members are inserted. For example, you may want to insert a member along with all its children, or insert only its children.

When you double-click a member that you already inserted in your report, by default its children are inserted after the parent member. You can control whether the children are inserted before or after, nested, or not inserted at all (**Tools**, **Options**, **Edit** tab).

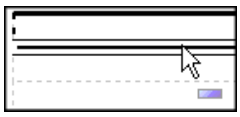
After you insert members, you may want to apply indentation to make the report easier to read.

Steps

1. From the toolbar, click the insert single member button and select how you want to insert members.

2. In the **Insertable Objects** pane, from the **Source** tab, drag the members that you want to the work area.

   - If you want to insert a member above or below another member, move your cursor above or below a call until a flashing black bar appears.

   - If you want to replace a member, move your cursor over a cell in the report until a flashing black box appears.

Create a Set of Members

Sets are used to group members that are logically related for various actions, such as formatting, nesting, and sorting. Creating sets is also useful when members may be dynamic over time, such as the child accounts of a total assets account, which may change from year to year. By creating a set, you do not have to modify the report each time accounts are added or removed.

When you create a set of members, you can insert other members before or after the set, but not within the set.

Steps

1. On the toolbar, click the set selection button to toggle between adding individual members and creating sets for members .

2. In the **Insertable Objects** pane, select the items that you want to include in the set and drag them to the work area.

Insert a Hierarchy

You can quickly insert entire hierarchies in a report using a single click of the mouse.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab, drag the hierarchy that you want to insert to the report.
2. In the **Insert Hierarchy** dialog box, choose which members to insert:
   - To insert only the root members of the hierarchy, click **Root members**.
   - To insert all members of the hierarchy, click **All members**.

     Tip: When inserting all members of a hierarchy into a crosstab, the **Level Indentation** property is set so that you can see the hierarchical structure when the report is run.

Insert a Member Property

You can insert member properties in a report. For example, an employee level may have a property named gender.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab, drag the member property that you want to the report.
2. If the **Insert Member Property** dialog box appears, choose whether to repeat or group the property:
   - To repeat the property in the rows or columns of the crosstab, click **Insert property only**.
   - To insert the property and have it grouped, click **Insert property and group it**.

     The property will appear once in the rows or columns of the crosstab.

Nest Members

When you insert members in your report, you may want to nest them in another row or column to make your report easier to use. You can nest members from different dimensions.

For example, in the following report, the quarters (Q1 to Q4) were selected from the Time dimension and nested only for the GO Americas member, which is from a different dimension.

Amount year to date		<#2004#>
<#GO Americas#>	<#Q1#>	<#1234#>
	<#Q2#>	<#1234#>
	<#Q3#>	<#1234#>
	<#Q4#>	<#1234#>
<#GO Consolidated corporate#>		<#1234#>
<#GO Consolidated eliminations#>		<#1234#>
<#GO Consolidated#>		<#1234#>
<#GO Asia Pacific#>		<#1234#>

You can also double-click nested members to insert their children, or expand them (p. 169).

Step

- In the **Insertable Objects** pane, from the **Source** tab , drag the members that you want to the work area.

  A flashing black bar indicates where you can drop an item

Search for Members

You can perform a member search to quickly find the data you want.

You can control the number of members that are returned in a search by specifying a value for the **Member display count limit (in source tree)** option (p. 57).

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , right-click a dimension, hierarchy, or level, and click **Search**.
2. In the **Words** box, type the words or characters you want to search for.
3. Click the search parameter you want to use.
4. If you want to search all descendents instead of just the immediate children, select the **Search all descendents** check box.

   For example, when performing a search in a hierarchy, selecting this check box returns members found in all the levels of the hierarchy.
5. Click **Search**.

Report Studio searches for all members for the object selected in step 1. The results appear in a hierarchical structure on the **Search** tab in the **Insertable Objects** pane. You can browse the hierarchy to explore members at lower levels.

Tip: You can insert members directly into a report. This can save you time, because you do not have to define a filter. For example, instead of inserting **Product line** from the **Source** tab and adding a filter for **Camping Equipment**, you can insert **Camping Equipment** from the **Search** tab.

Filter Data

Add a filter expression to focus a report and minimize processing time by excluding unwanted data. For example, you can filter data to show customers who placed purchase orders valued at over one thousand dollars during the past year. When you run the report, you see only the filtered data.

If an expression is used in multiple reports, or by different report authors, ask your modeler to create the expression as a standalone object in the model and include it in the relevant package.

For information about creating filters in the package, see the Framework Manager *User Guide*.

If you are working with a dimensional data source, you can also define slicers (p. 174). If you are filtering non-numeric data, you must select members from the data tree instead of typing member names (p. 359). If you are filtering a multiple-fact query, you should understand the differences between using a detail filter or summary filter (p. 215).

Steps

1. To add a filter that was created in the package, in the **Insertable Objects** pane, on the **Source** tab , drag the filter you want to the report.

   You must drag the filter to an object that is linked to a query. If there is more than one query defined in the report, drag the filter to an object linked to the query you want.

2. From the **Data** menu, click **Filters** .

   If you have more than one query defined in the report, you must first click an object linked to a query. If there is more than one query defined in the report, click an object that is linked to the query you want.

   Tip: You can also define filters in Query Explorer (p. 213).

3. In the **Filters** dialog box, decide what type of filter you want to create:
   - To add a filter that will apply to detail values, click the **Detail Filters** tab.
   - To add a filter that will apply to summary values, click the **Summary Filters** tab.

4. Click the add button .

5. In the **Available Components** box, define the filter expression:
   - To include data items not shown in the report, on the **Source** tab, double-click data items.
   - To include data items that are in the report but not necessarily in the model (such as calculations), on the **Data Items** tab , double-click data items.
   - To include data items from a specific query (p. 213) in the report, on the queries tab, double-click data items.
   - To add functions, summaries, and operators to the filter expression, on the **Functions** tab, double-click elements.

Note: You can insert only functions that return a boolean value. For example, you cannot insert the function **topCount** because it returns a set of data. Filter expressions must resolve to a boolean in order to be valid.

- To include a value that is derived from a parameter, on the **Parameters** tab, double-click a parameter.

  Parameters are used to define prompts (p. 199), drill-through reports (p. 273), and master detail relationships (p. 282).

Tip: You can also type the filter expression directly in the **Expression Definition** box. When typing date values, the date must be in the YYYY-MM-DD format.

For more information about creating expressions, see "Using the Expression Editor" (p. 357).

6. Click the validate expression button and click **OK**.
7. In the **Usage** box, specify whether the filter is required, optional, or not to be used.
8. If you created a detail filter, in the **Application** box, click one of the following options:
   - To apply the filter to individual records in the data source, click **Before auto aggregation**.

     For example, you want to filter out individual orders of a specific product type within a product line that generated less than one thousand dollars in revenue.

   - To apply the filter to data after the query has grouped and summarized at the lowest level of detail, click **After auto aggregation**.

     For example, you want to filter out product types within a product line that generated less than ten thousand dollars in revenue.

   For example, you have a list that contains the data items Product line, Product type, and Revenue. The revenue values you see are aggregated to the product type level. If you create detail filter on Revenue and you choose to filter values before auto aggregation, you are filtering non-aggregated revenue values. If you choose to filter values after auto aggregation, you are filtering revenue values aggregated to the product type level.

   Tip: If you are using a dimensional data source, after aggregation is always applied, because by definition, dimensional data is aggregated.
9. If you are filtering a summary, click the ellipsis (...) button under **Scope** and select the grouping level for which you want to apply the filter.

   For example, a revenue report is grouped on product line and product type. You can choose to filter total revenue for either the product line or product type level.

   If the query in which you are creating the summary filter is linked to more than one data container, the data item that you select as the grouping level must be grouped in all data containers linked to the query. Otherwise, the report will not run successfully.

   Tip: To filter at the overall level, do not click a level. For example, if a report has an overall footer that shows the total revenue for all products, by not choosing a level you will apply the filter to the footer. In addition, if you are using a dimensional data source, excluding a parent level excludes its children, and excluding all children excludes the parent.

10. Click **OK**.

Filter Values Using Context

When working with dimensional data, you can use context filters, also known as slicer filters, to quickly focus your report on a particular view of the data.

For example, you create a crosstab with accounts in the rows, years in the columns, and expenses as the measure. To change the context to Asia, you drag Asia from the source tree to the **Context filter** section of the overview area. The crosstab then shows only the values for Asia. Changing context changes the values that appear. It does not limit or change the items in the rows or columns.

The data items that are used as the context filter appear in the report header when you run the report.

You can create multiple context filters or slicers if you want to filter across two or more dimensions. You cannot create multiple context filters against the same dimension.

If you want to use a calculation or expression to define a slicer, create a slicer member set using the query explorer. A slicer member set is an expression that returns a set of members from the same dimension. For example, you can define a slicer member set that filters for the top 5 products with the greatest revenue.

Slicer filters are different from other filters. When you filter data, members that do not meet the filter criteria are removed from the report. A slicer does not remove members from a report. Instead, their values are removed, and you see blank cells. For example, if you have a crosstab with Years and Quarters as rows and Revenue as columns and you define a slicer member set that consists of the first two quarters in 2006, you get the following result:

		Revenue
2004	Q1	
	Q2	
	Q3	
	Q4	
2005	Q1	
	Q2	
	Q3	
	Q4	
2006	Q1	344,124,267.07
	Q2	391,874,462.51
	Q3	
	Q4	
2007	Q1	
	Q2	
	Q3	

In addition, any summary values in the report are recomputed to reflect the results returned by the slicer.

You can also use context filters as prompts when the report is run (p. 200).

Steps to Create a Simple Context Filter

1. In the **Insertable Objects** pane, from the **Source** tab, select or search for one or more items to filter on.
2. Drag the item that you want to filter on into the **Context filter** section of the overview area.
3. To change context, select a new item from the drop-down list under **Context filter**.

Steps to Create an Advanced Slicer Filter

1. Pause the pointer over the query explorer button and click the query to which you want to add a slicer.
2. On the **Toolbox** tab, drag **Slicer Member Set** to the **Slicer** pane.
3. Drag the members that you want to the **Expression Definition** box.

   You must join the members in the expression by using the `set` function. For example, the following expression contains the Fax and Telephone members from the Order Method dimension:

   `set`([Fax],[Telephone])
4. To add members from another dimension, repeat step 2 to create a separate slicer member set.

Filtering Data Using an SAP BW Data Source

If you are working with an SAP BW data source, you must consider additional things when applying filters.

If you apply a filter and a sort to an item that corresponds to the leaf-level of a recursive hierarchy, siblings may appear in the report, although the aggregated values are correct. Siblings are characteristic values with the same parent of the filtered member.

Each level in an SAP BW hierarchy has an item with the same name as the level, and has a role of _businessKey. Such items are known as level identifiers. The level identifier must be an exact value, for the operators =, <, and > to work. For example, for the filter [Office] > 'Chicago' to work, the value 'Chicago' must exist in the data source. If you do not know the exact values, you can apply the filter to one of the attribute items associated with the level, such as [OfficeLongName] > 'C'. Filters on non-identifiers are possible, but they are slower because SAP BW is optimized for queries based on level identifiers.

When filtering time-related data, only the level identifier items of the time-related characteristics in SAP BW, such as 0CALDAY and 0CALMONTH, should be used for performing anything other than equality filters. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

Ignoring the Time Component in Date Columns

Database systems use Date, Time, and Timestamp to represent date and time values. While a timestamp type holds a date and time component, an application may allow the RDBMS to default the time component. When rows are inserted, updated, or queried, the application may specify only a date value and leave the RDBMS to extend the value to include a default time (usually 00:00:00.000).

The challenge with a timestamp is when the application has no immediate interest in the time component. For example, the business question "How many orders were taken today?" implies all orders taken irrespective of what time in the day the order was booked. If the application defaulted the time component as it stored rows, the query used to answer the question returns the count of orders taken today. If the application stored the actual time component, the query likely returns no data, because the number of orders entered at midnight is probably zero.

Relying on dates defaulting the time can be dangerous if the application changes and starts to capture actual times. To avoid this problem, you can

- truncate the time by creating a derived column
- convert the timestamp to a date
- create a hi-low filter

Truncating the Time by Creating a Derived Column

In the Framework Manager model, create a derived column using a calculated expression that truncates the time from a timestamp, and returns a timestamp containing the original date and a default time. For example, if you use an Oracle data source, the following expression creates the derived column DATEONLY from COL1, where COL1 contains the values as stored by the application while DATEONLY contains the dates with the default time of 12:00:00 AM:

```
Select COL1,trunc(COL1) as DATEONLY from [SCOTT_TIGER].DATES
```

Tip: You can change the data format of the column to show only the date value by specifying **Short** for the **Date Style** property.

You can then apply filters to the DATEONLY column that would return the correct results. If you create a parameter in Report Studio that filters on this column, the default prompt will present a date and time control because the data type is still a timestamp.

Converting the Timestamp to a Date

In the Framework Manager model, define a calculation that uses the CAST function to convert the timestamp to a date. For example, the following expression converts the data type of the column COL1 to date:

```
cast ([SCOTT_TIGER].[DATES].[COL1],DATE)
```

If you create a parameter in Report Studio that filters on this calculation, the default prompt presents a date control.

Ignoring the Time by Creating a Hi-Low Filter

You can create a filter to ignore the time. In the Framework Manager model, create a filter in the form date-column between date-lowtime and date-hightime. For example, the following expression returns all values between 00:00:00:000 and 23:59:59:000 for a given day:

```
[SCOTT_TIGER].[DATES].[COL1] between
?p1? and cast(substring(?p1?,1,10),'23:59.59.000',timestamp)
```

Sort Data

By default, Report Studio retrieves items in the order defined in the data source. You can sort items to quickly view the most important data. Report Studio provides different sorting options depending on whether you use relational or dimensional data and depending on what type of data item you select.

Layout Sorting

If you use relational data, you can sort items in a list in ascending or descending order based on a value or a caption, such as revenue or employee name. You can also perform advanced sorting (p. 179) to sort columns within groups or to sort a row or column using another data item.

This type of sorting is performed in the report layout and not in the query. Sorting in the layout is useful when working with lists, when your report includes relational data, and when working with expression-based data items.

Set Sorting

If you use dimensional data, you can sort a set of members based on a caption (heading label). You can also perform advanced sorting (p. 179) to sort sets using an intersection (tuple) or using a member property. By default, when you select **Sort Set - Ascending** or **Sort Set - Descending**, sets are sorted based on captions.

For example, a set contains regions and you sort the regions within the set in descending alphabetical order. Only extended data items that represent sets support set sorting.

This type of sorting is performed in the query.

Before you can perform set sorting, you must create sets for the members in your report (p. 169).

You cannot sort data items from different dimensions that are intermixed. For example, you cannot sort on Data_Item1_Dimension1, Data_Item1_Dimension2, Data_Item2_Dimension1.

You cannot apply set sorting and layout sorting at the same time. Sorting applied to an item will overwrite any previous sorting that was applied.

Sorting by Value

If you use dimensional data, you can sort members in the sets on the opposite axis based on the value of a member or measure that you select.

For example, a crosstab contains years inserted as individual members in the columns, product lines inserted as a member set in the rows, and revenue as the measure. You select 2004 and click the

sort by value button and select to sort in descending order. The values in the 2004 column are sorted.

Revenue	2004	2005
Personal Accessories	391,647,093.61	456,323,355.9
Camping Equipment	332,986,338.06	402,757,573.17
Golf Equipment	153,553,850.98	168,006,427.07
Outdoor Protection	36,165,521.07	25,008,574.08
Mountaineering Equipment		107,099,659.94

This type of sorting is performed in the query.

Before you can perform this sorting, you must create sets for the members in your report (p. 169).

In nested crosstabs, items are sorted based on the values of the innermost nested row or column of the opposite axis. For example, a crosstab contains years in the columns and retailers nested within product line in the rows. Select 2004, sort in descending order, and you see retailers listed from the largest value to the smallest value for each product line.

Considerations When Sorting SAP BW Data

If you use a SAP BW data source, each level in a hierarchy has an item with the same name as the level, and has a role of _businessKey. Such items are known as level identifiers. You should use only the level identifier of all the time-related characteristics, such as 0CALDAY and 0CALMONTH, for sorting. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

Tip: By level identifier, we are referring to a query item with the same name as the level that has the role of _businessKey. Each level in a SAP BW hierarchy has a level identifier.

Steps to Sort in the Layout

1. Click the data item on which you want to sort.
2. From the **Data** menu , click **Sort Ascending** or **Sort Descending**.

   Tip: If you are sorting an extended data item, click **Layout** first.

   An up arrow or down arrow appears beside the data item to indicate that a sort order was set.

   When you specify a sort order for more than one column, the columns are sorted in the order that they were inserted in the report. For example, you add columns A, B, and C to a report and specify a sort order for each. When you run the report, column A is sorted first, followed by B and then C. You can change the order in which the columns are sorted.

   Tip: To remove a sort order, click **Don't Sort**.

Steps to Sort Sets by Caption

1. In a crosstab, select a set to sort.
2. Click the sort button on the toolbar and click **Sort Set - Ascending** or **Sort Set - Descending**.

Tip: To remove a sort order, click **Don't Sort**.

Steps to Sort Sets in the Opposite Axis by Value

1. In a crosstab, a member or measure to sort.

   For example, to sort a set in the row axis, select a member or measure in the column axis.

2. Click the sort opposite axis sets by value button on the toolbar.

   Tip: To remove a sort order, click **Don't Sort**.

Perform Advanced Sorting

You can perform advanced sorting in a report. If you use relational data, in a list, you can sort columns within groups and change the sort order of columns. In a crosstab, you can sort a row or column by another item, such as Order year by Revenue.

If you use dimensional data, you can sort using the property of a member. For example, your report includes employee names and you want to sort the employees using the gender property. You can also sort using an intersection of members, also know as a tuple. For example, you want to sort the employees using the value of sick days taken for the year 2006 (p. 191).

Steps to Sort in the Layout

1. Click a column.

   In a crosstab, click the row or column for which you want to perform advanced sorting.

2. From the **Data** menu, click **Sort**, **Advanced Sorting**.

3. If you are sorting a list and you want to sort a column within a group, do the following:

   - In the **Groups** pane, under the **Groups** folder, expand the folder of the grouped column (p. 80) that you want.
   - In the **Data Items** pane, drag the data items you want to sort to the **Sort List** folder.

     Tip: You can also drag data items from the **Detail Sort List** folder.
   - Click the sort order button to specify ascending or descending order.

4. If you are sorting a list and you want to change the sort order of columns, in the **Groups** pane, change the order of columns in the **Sort List** folder of a group, or in the **Detail Sort List** folder.

   Tip: Add items to the **Detail Sort List** folder to sort items that are not groups.

5. If you are sorting a crosstab, do the following:

   - From the **Data Items** pane, drag the item you want to sort on to the **Sort List** pane.

     For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. To sort Order year by Revenue, drag Revenue to the **Sort List** pane.

- Click the sort order button to specify ascending or descending order.

Steps to Sort Sets

1. Select a set, member, or measure to sort.
2. Click the sorting button on the toolbar and click **Advanced Set Sorting** or **Layout**, **Advanced Sorting**.
3. Specify the sorting options that you want.
4. If you are sorting members from different levels and want to preserve the hierarchy, select the **Hierarchized sorting** check box.
5. If you want to sort items using a member property, under **Sort by**, click **Property** and click the ellipsis (...) button. Then, navigate the tree and select the member property you want to use.
6. If you want to sort items using an intersection of members, or tuple, under **Sort by**, click **Intersection (tuple)** and click the ellipsis (...) button. Then, from **Available members and measures**, select the items you want to use and click the right arrow to move them to the **Intersection members and measures** list.

Create Sections

Create sections in a report to show a data item as the heading of a section. When you run the report, separate sections appear for each value.

Creating sections is similar to creating headers by grouping on a data item (p. 80). The difference is that section headers appear outside the list, crosstab, chart, or repeater. In addition, you can group data items only in lists.

When working with dimensional data, you can also create page layers to show values on a separate page for each member (p. 181).

Steps

1. Click the column that you want to show as a section heading.
2. Do one of the following:
   - From the **Structure** menu, click **Section** .

     If the column is in a list or repeater, this menu option will create sections without creating a master detail relationship. This can improve performance when running the report.

     If the column is in a crosstab or chart, this menu option will create sections using a master detail relationship.
   - From the **Structure** menu, click **Section using Master/Detail**.

     This menu option creates sections using a master detail relationship.

Sections are created, and a section header appears. The data container that contains the column used to create sections is embedded in a list.

3. To add or remove section headers and footers, from the **Structure** menu, click **List Headers & Footers**, select or clear the appropriate check boxes, and click **OK**.

   The section header and footer check boxes appear under **Overall header**.

Remove Sections

To remove sections and reinsert the data item used to create the sections to the data container, do the following:

Steps

1. From the **View** menu, click **Page Structure**.
2. Expand the page containing the data container with the section.
3. Expand **Page Body** until you see the data container in which you added the section.

   The data container is nested in the **List Column Body** object of the **List** object that is created when you add a section.
4. Drag the data container to **Page Body**.

   The data container appears as a node of **Page Body**.
5. Delete the **List** object.
6. From the **View** menu, click **Page Design**.
7. In the **Insertable Objects** pane, click the **Data Items** tab .
8. Drag the data item that was used as a section header back into the data container.
9. If you created sections in a list without creating a master detail relationship, click the data item and then click the group/ungroup button to ungroup the item.

For information about the page structure view, see "The Page Structure View" (p. 134).

Create Page Layers

When working with dimensional data, you can create page layers in a report to show values on a separate page for each member. For example, your report contains payroll information for the entire company. You want to filter your report so that values for each department appear on a separate page.

After you create page layers, a caption appears in the header to indicate the contents of each page. You can navigate between the different pages using links below the report.

Creating page layers is similar to filtering using context (p. 174). However, with context filters, values are filtered according to the member you add to the **Context filter** area. With page layers, the report is split into a separate page for each child of the member you add to the **Page layers** area.

If you want to create more complex page layers in your reports, such as a report book with title and end pages, create page sets (p. 152).

Steps

1. In the source tree, select or search for one or more items to filter on.
2. Drag the item that you want to filter on into the **Page layers** section of the overview area.

   The crosstab shows the results for the children of the selected item on separate pages and a drop-down list appears under **Page layers**.

   Tip: To navigate between pages, click the **Page down** and **Page up** links below the report.
3. If you want to replace the page breaks with items from the same dimension, select an item from the drop-down list. If you want to delete the page breaks, from the drop-down list, click **Delete**. If you want to delete all the page breaks, right-click the **Page layers** area and click **Delete All**.

Summarizing Data

In a report, there can be detail and summary aggregation. The detail aggregation specifies how a data item is totaled at the lowest level in a report. In lists, the detail aggregation specifies how the values that appear in the rows are totaled. In crosstabs, the detail aggregation specifies how the values in the cells are totaled. For example, the detail aggregation for a measure like Revenue might be Total in both lists and crosstabs. In the following list report, this means that the values you see for the Revenue column represent the total revenue for each product type.

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	$5,401,132.08
	Tents	$47,456,718.10
	Sleeping Bags	$11,688,944.32
	Packs	$14,627,996.26
	Lanterns	$10,539,200.16
Golf Equipment	Putters	$2,310,367.24
	Golf Accessories	$811,516.72
	Irons	$8,801,464.98
	Woods	$13,982,116.64

The summary aggregate specifies how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab.

Revenue	2004	2005	2006	Total(Order year)
Camping Equipment	$20,471,328.88	$31,373,606.46	$37,869,055.58	**$89,713,990.92**
Golf Equipment	$5,597,980.86	$9,598,268.88	$10,709,215.84	**$25,905,465.58**
Mountaineering Equipment		$9,642,674.54	$11,248,676.06	**$20,891,350.60**
Outdoor Protection	$1,536,456.24	$988,230.64	$646,428.04	**$3,171,114.92**
Personal Accessories	$7,144,797.52	$10,955,708.04	$13,793,960.30	**$31,894,465.86**
Total(Product line)	**$34,750,563.50**	**$62,558,488.56**	**$74,267,335.82**	**$171,576,387.88**

You can specify detail and summary aggregation in different ways by using any of the following:

- aggregation properties that are specified in the model (p. 183).
- the **Auto Group & Summarize** property (p. 183).

- the aggregate button in the toolbar (p. 184).
- aggregation properties for a data item (p. 185).
- aggregate functions in data item expressions (p. 185).
- the aggregation mode (p. 186).
- the solve order (p. 191).

If a summary is applied to a report that contains binary large object (BLOB) data, such as images or multimedia objects, you cannot also perform grouping or sorting.

OLAP Data Sources

Summary options depend on the data source type that you are using. If you are querying an OLAP data source, all measure values that appear in reports are pre-summarized because the data source contains rolled up values. The type of aggregate that is used is specified in the data source itself. As a result, you cannot specify a different detail aggregation in a report. The report always renders the type of aggregate that is specified in the data source.

The Type of Data

How data is aggregated also depends on the type of data that you are aggregating. Aggregation rules are applied differently to facts, identifiers, and attributes. For example, if you aggregate a data item that represents part numbers, the only aggregate rules that apply are count, count distinct, count non-zero, maximum, and minimum. For information about how to determine the type of data a data item represents, see "Add Data to a Report" (p. 64).

Use the Aggregation Properties Set in the Model to Specify Detail or Summary Aggregation

When working with relational or dimensionally modeled relational (DMR) data sources, you can use the aggregation properties specified for the query item in the Framework Manager model instead of specifying detail or summary aggregation in the report. The model specifies the default summary function for each query item.

Aggregation functions are mapped between data sources, Framework Manager, and Report Studio (p. 355).

Set the Auto Group & Summarize Property

Set the query property **Auto Group & Summarize** to specify whether Report Studio should group non-fact data items (identifiers and attributes) and apply aggregate functions to aggregate fact data items in lists.

If you are using an OLAP data source, data is always summarized regardless of how this property is set.

Steps

1. Pause the pointer over the query explorer button and click the query that you want.
2. In the **Properties** pane, set the **Auto Group & Summarize** property to the value that you want:
   - To group non-aggregate fact data items and apply aggregate functions to aggregate fact data items in lists, set this property to **Yes**.
   - To render detail rows, set this property to **No**.

Use the Aggregate Button to Add Simple Summaries

You can add simple summaries to the groups in a report by using the aggregate button in the toolbar. This button provides a subset of the aggregate functions available in Report Studio. In addition, a **Custom** option is available so that you can add your own aggregate function in the expression of the data item (p. 185).

The aggregate button sets the rollup aggregate property (p. 185) for the data item to the selected summary aggregate, and places the data item into an appropriate footer. A footer is created for each group as well as an overall footer, unless they already exist.

Steps

1. Open the report that you want.
2. Click the column to which you want to add a summary.
3. From the **Data** menu, click the kind of summary that you want (p. 351).
4. If you want to change the summary label, do the following:
   - Click the label.
   - In the **Properties** pane, click the **Source Type** property and then click the source type that you want to use to define the label.

     For example, click **Data Item Value** to produce a dynamic label for the summary based on data item values.
   - Depending on the source type you chose in the previous step, click the property below **Source Type** and specify the label.

     For example, if clicked **Data Item Value** as the source type, click the **Data Item Value** property and click the data item that you want to use to define the label.

In lists, the summary appears as a footer (p. 135). If the column to which you added a summary is grouped, group and overall summaries appear. In crosstabs and charts, the summary appears as a node (p. 86).

Tips

- If you want to change a summary, click it, and in the **Properties** pane, under **Data Item**, click **Rollup Aggregate Function** (p. 185) and choose a different function.
- In crosstabs, you can add multiple summaries at the same level. For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. For Product line, you can add the **Total** summary as a header, which will total all revenue for each order year. You can then add the **Average** summary as a footer, which will give the average revenue of all product lines for each order year.

Specify Aggregation Properties for a Data Item

For each data item in a report, you can specify detail and summary aggregation properties so that you can easily manage summaries without having to create complex data item expressions (p. 185).

Steps

1. Click the data item for which you want to set detail or summary aggregation.
2. In the **Properties** pane, set the **Aggregate Function** or the **Rollup Aggregate Function** property to the function that you want (p. 351).

Use Aggregate Functions in Data Item Expressions

You can use aggregate functions in data item expressions. The aggregate functions in the Expression Editor (p. 357) that have the same name as the aggregate functions available through the **Aggregate Function** and **Rollup Aggregate Function** properties (p. 185) operate the same way. For example, setting the **Rollup Aggregate Function** property to **Total** is the same as changing the expression of the data item to

total([Revenue])

In general, report maintenance is easier if the **Aggregate Function** and **Rollup Aggregate Function** properties are used rather than adding aggregate functions to data item expressions. Use summary functions in expressions if the required summary is not supported as an aggregate or rollup aggregate property or if the complexity of the expression cannot be managed using the data item properties. Add an aggregate function to a data item expression if one of the following conditions applies:

- The underlying data source is relational and you want to use database vendor-specific aggregate functions.
- You want to use aggregate functions that require more than one parameter, such as percentile.
- You require aggregate expressions that are not available in the aggregation properties, such as a `for` clause.

For example, your report uses Product line, Product type, and Quantity. You want to calculate the percentage of the parent quantity represented by each product type. This requires you to compare the total quantity for each product type to the total quantity for the parent product line. In other

words, your expression requires aggregates at different group levels. You use aggregate functions in the data item expression with a `for` clause to specify the group level, as follows:

total([Quantity] for [Product type]) / total([Quantity] for [Product line])

Steps

1. Click the data item that you want.
2. In the **Properties** pane, double-click the **Expression** property.
3. Add the aggregate functions that you want to the expression for the data item.

Specify the Aggregation Mode

If you are working with a dimensional or a dimensionally modeled relational (DMR) data source, you must specify the aggregation mode that you want Report Studio to use when aggregating data in crosstabs and charts (p. 186). The aggregation mode sets the aggregation clause to be used in the data item expression.

Steps

1. From the **Tools** menu, click **Options**.
2. Click the **Edit** tab.
3. Click **Aggregation mode** and select the aggregation mode (p. 57) that you want to use.
4. Click **OK**.

Aggregating Values in Crosstabs and Charts

In crosstabs and charts, aggregated values are calculated using one of the following `aggregate` expressions:

- `aggregate(`*`measure`* `within set` *`set expression`*`)`

  Aggregates the member values from the data source within the current content.

- `aggregate(`*`measure`* `within detail` *`data item`*`)`

  Aggregates the lowest level of details in the report.

- `aggregate(`*`measure`* `within aggregate` *`data item`*`)`

  Aggregates each level of details in the report.

You decide which `aggregate` expression is used by setting the aggregation mode (p. 186).

For example, in the following crosstab, if you specified **Total** as the summary, these aggregate expressions are produced for each aggregation mode:

		<#Revenue#>	<#Revenue#>
<#Quarter#>	<#Month#>	<#1234#>	<#1234#>
	<#Month#>	<#1234#>	<#1234#>
<#Total(Quarter)#>		**<#1234#>**	**<#1234#>**

- Total ([Revenue] within set [Quarter])

  This expression totals the quarter values from the data source at the intersecting product line. At the bottom right corner, it totals the aggregate over all product lines for each quarter.

- Total ([Revenue] within detail [Quarter])

  This expression totals the month values visible in the report at the intersecting product line. At the bottom right corner, it totals all of the intersecting month - product line values visible in the report.

- Total ([Revenue] within aggregate [Quarter])

  This expression totals the month values visible in the report at the intersecting product line into quarters. At the bottom right corner, it does the same, but starting with the aggregate over all product lines for each month.

In simple cases, the members and values visible in the report and the aggregate rules in the report are the same as those in the data source, and all of these expressions produce the same results.

For example, for the quarter and month values, if you are totaling the values for all months in all quarters, it makes no difference whether the visible values, the values in the cube, or month and quarter values are used. The result is the same.

		Revenue
2005 Q 3	2005/Sep	$5,300,599.08
2005 Q 4	2005/Oct	$11,778,348.92
	2005/Nov	$3,353,399.48
	2005/Dec	$4,645,069.28
2006 Q 1	2006/Jan	$3,119,227.22
	2006/Feb	$3,045,478.88
	2006/Mar	$6,867,919.18
2006 Q 2	2006/Apr	$8,306,586.26
	2006/May	$5,129,112.24
	2006/Jun	$6,927,447.24
2006 Q 3	2006/Jul	$5,078,789.96
	2006/Aug	$5,180,166.38
	2006/Sep	$4,892,742.38
2006 Q 4	2006/Oct	$15,887,237.28
	2006/Nov	$3,609,599.42
	2006/Dec	$6,240,652.88
Total(Quarter)		**$171,576,387.88**

Different results appear when you start filtering, changing aggregation types, or use set expressions or unions.

For example, the following crosstab shows the quantity of products sold across all product types for each product line. The bottom of the crosstab has three summary values that show the average quantity of products sold by product line.

		Quantity
Camping Equipment	Cooking Gear	13,400,351
	Tents	3,164,285
	Sleeping Bags	3,153,218
	Packs	2,756,540
	Lanterns	4,826,755
Mountaineering Equipment	Rope	418,560
	Safety	1,135,194
	Climbing Accessories	5,850,251
	Tools	2,496,086
Personal Accessories	Watches	4,299,195
	Eyewear	20,311,396
	Knives	7,229,314
	Binoculars	1,075,087
	Navigation	1,992,713
Outdoor Protection	Insect Repellents	5,800,964
	Sunscreen	5,384,127
	First Aid	829,354
Golf Equipment	Irons	391,445
	Woods	317,939
	Putters	1,284,570
	Golf Accessories	3,119,747
Average(Product line) - within detail		**4,249,385.29**
Average(Product line) - within aggregate		**4,040,006.76**
Average(Product line) - within set		**17,847,418.2**

Each summary value uses a different aggregation mode that is indicated in the summary name.

- **Average(Product line) - within detail**

  This summary is the average of the detail values in the crosstab.

- **Average(Product line) - within aggregate**

  For each product line, the average of the detail values is calculated. This is equivalent to applying the `average` aggregation function to the Product type column. This summary is the average of the product type averages.

- **Average(Product line) - within set**

  This summary is the average of the product type values rolled up into sets at the Product line level. The values are obtained from the data source. If filters or slicers exist, the values are recomputed using the aggregation rules defined in the data source.

In most cases, you should use the `within detail` aggregation mode because the results are easiest to understand, and they are the same as for footers in a grouped list report. In more complex cases, you may consider the `within aggregate` aggregation mode. The `within set` aggregation mode should be reserved for reports with a purely dimensional focus. For example, when there are no detail or summary filters defined in the report.

Aggregating Member Sets

When you work with dimensional data sources and you aggregate member sets, if an explicit summary function such as Total is used and the set contains duplicates, the result is double-counted.

If you are using an OLAP data source, the result produced for the Automatic summary function depends on the data source.

For example, the product line rows below were defined using the expression

`union`([Product line], [Camping Equipment], `ALL`)

where [Product line] is the level.

	Revenue
Camping Equipment	*1,589,036,664.03*
Mountaineering Equipment	*409,660,132.9*
Personal Accessories	*1,885,673,307.78*
Outdoor Protection	*75,994,296.25*
Golf Equipment	*726,411,367.89*
Camping Equipment	*1,589,036,664.03*
Aggregate(Product line)	**4,686,775,768.85**
Total(Product line)	**6,275,812,432.88**

The summary **Aggregate(Product line)** is the sum of the product lines excluding duplicates. For more information about how the Aggregate function is processed, see "Summary Functions" (p. 351).

Create a Calculation

Create a calculated column to make a report more meaningful by deriving additional information from the data source. For example, you create an invoice report, and you want to see the total sale amount for each product ordered. Create a calculated column that multiplies the product price by the quantity ordered.

If an expression is used in multiple reports, or by different report authors, ask your modeler to create the expression as a standalone object in the model and include it in the relevant package.

If you want to perform a more complex calculation, using expressions and functions, insert a calculated member (p. 190).

You can add calculations to lists, crosstabs, and all other data containers. You can also add calculations directly to a page header, body, or footer. However, you must first associate a query to the page. For more information, see "Add a Page" (p. 151).

Cognos does not currently support queries with both dimensional functions and running totals. Running totals are supported only for the overall level of a dimension.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab .
2. To create a calculation that will be applied to data, do the following:
   - Drag **Query Calculation** to the report.

     The **Create Calculation** dialog box appears.
   - In the **Name** box, type a name for the calculation and click **OK**.

3. To create a calculation in the layout that contains run-time information, such as current date, current time, and user name, drag **Layout Calculation** to the report.
4. In the **Available Components** box, define the calculation:
   - To include data items not shown in the report, on the **Source** tab, double-click data items.
   - To include data items that are in the report but not necessarily in the model, such as calculations, on the **Data Items** tab, double-click data items.
   - To include data items from a specific query (p. 213) in the report, on the queries tab, double-click data items.
   - To add functions, summaries, and operators to the filter expression, on the **Functions** tab, double-click elements.
   - To include a value that is derived from a parameter, on the **Parameters** tab, double-click a parameter.

     Parameters are used to define prompts (p. 199), drill-through reports (p. 273), and master detail relationships (p. 282).

     Tip: You can also type the expression directly in the **Expression Definition** box.

     For more information about creating expressions, see "Using the Expression Editor" (p. 357).
   - To include data items not shown in the report, on the **Source** tab, double-click data items.
5. Click the validate expression button and click **OK**.

Insert a Calculated Member or Measure

When working with dimensional data, insert a calculated member or measure into your report to add a new row or column with values based on a calculation.

You use the expression editor to define the calculation.

You can also create a calculation that uses an intersection (tuple) that you already defined (p. 191).

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag **Calculated Member** or **Calculated Measure** to the report.
3. In the **Name** box, type a name for the calculation.
4. In the **Calculation hierarchy** box, select the hierarchy within which this calculated member or measure is to be created and click **OK**.

   The **Data Item Expression** dialog box appears
5. In the **Available Components** box, define the calculation:

- If you want to include data items not shown in the report, on the **Source** tab, double-click the data items.
- If you want to include data items that are in the report but not necessarily in the model, such as calculations, on the **Data Items** tab, double-click the data items.
- If you want to add functions, summaries, and operators to the filter expression, on the **Functions** tab, double-click them.

6. Click the validate expression button and click **OK**.

Create an Intersection (Tuple)

When working with dimensional data, an intersection, also known as a tuple, is useful for obtaining a value from the combination of two or more members that you specify. Each member must be from a different dimension.

You can insert an intersection to create a complex calculation without using expressions. You can isolate the intersection of dimensional selections to a single row or column, and then use the intersection in other calculations (p. 190).

An intersection appears as a single unified member instead of its component members. By combining members with measures or with time-related members, you can create an intersection that reveals important information about your business.

You can also use intersections to sort reports.

By default, when you add an intersection in your report, its label is a collection of its members, such as 2004, Web, and Cooking Gear. You can change the name of the label.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag the **Intersection** (**Tuple**) object to the report.
3. From **Available members and measures**, select the items you want to use and click the right arrow to move them to the **Intersection members and measures** list.

Multiple Calculations in Crosstabs and Charts

An intersection point in a crosstab or chart can contain a value that is derived from multiple calculations.

If the query expressions for the row and for the column that intersect in a crosstab both include calculations, the intersecting value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. If you change the order in which the calculations are performed, the resulting value at the intersection point changes.

The solve order is a positive numeric value that you assign to data items. Negative values are not supported. The data item with the lowest value is calculated first, followed by the next higher value,

and so on. The data item with the highest solve order is computed last. If you do not specify a solve order, the default value is 0. In crosstabs, if more than one data item has the same solve order, column items are calculated first and then row items.

For example, a crosstab contains the columns Actual revenue, Sales target, and a third column that calculates the percentage by which actual revenue exceeds target revenue. The crosstab contains a row for each Sales year and a summary row that calculates the overall totals for each of the three columns. The cell where the percentage calculation and the overall total calculation intersect contains only one value. By default, Cognos 8 calculates the column percentage first, and the summary row is calculated last. The value that appears at the intersection is therefore a sum of the percentages, which is not a meaningful result. The value is more meaningful if the overall percentage is calculated after the actual and sales target revenues are summed. Specifying a solve order for the percentage calculation that is higher than the overall total calculation gives the following result:

	Actual revenue	Sales target	*Percentage by which Actual Exceeds Target*
2004	$43,025,310.66	$15,051,551.00	*186%*
2005	$86,096,447.76	$29,408,080.00	*193%*
2006	$102,253,419.64	$36,068,756.00	*183%*
Total(Sales year)	**$231,375,178.06**	**$80,528,387.00**	**187%**

It is not necessary to specify the solve order for the overall total calculation. Because the default solve order is 0, setting the solve order for the percentage calculation to a value higher than 0 is sufficient.

Nested Calculations in Crosstabs

If a crosstab contains nested calculations, the calculations are performed in the following order by default:

- outermost calculation on the columns
- innermost calculation on the columns
- outermost calculation on the rows
- innermost calculation on the rows

If there are solve order ties, column items are calculated before row items, and outer nesting items before inner nesting items.

Resolve Multiple Calculations for Crosstabs and Charts

If an intersection point in a crosstab or chart contains a value that is derived from multiple calculations (p. 191), you can specify the order in which to perform the calculations. The solve order must be a positive numeric value. The calculations are performed from lowest to highest solve order value.

Setting the **Rollup Aggregation Function** property to **Calculated** (p. 351) for a data item assigns a solve order that is higher than any other data item in a query, unless a solve order is explicitly

specified. If there are multiple data items that have the **Rollup Aggregation Function** property set to **Calculated**, the normal rules for solve order precedence apply.

Steps

1. Click the data item for which you want to specify the solve order.

   A data item can have only one solve order value. Consequently, if a data item is in a query that is used in more than one data container, and you want the data item to have a different solve order in each container, you must use a separate data item.

   Tip: In crosstabs, to specify the solve order for the default measure (p. 198), click the crosstab and, in the **Properties** pane, click the select ancestor button and click **Crosstab**.

2. In the **Properties** pane, type an integer value in the **Solve Order** property.

   To specify the solve order for the default measure, click **Default Measure Solve Order**.

Resolve Multiple Calculations when Querying Dimensional Data Sources

Dimensional data sources may include data source calculations. Because Cognos 8 cannot predict the presence of such calculations, specifying the **Solve Order** property may be unable to resolve the solve order correctly. You can suppress potentially meaningless results when querying dimensional data sources that include data source calculations.

When a query-defined calculated member and a data source calculated member intersect, the value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. Whether the final value is meaningful depends on the order in which the calculations are performed. Because Cognos 8 cannot determine the presence of data source calculated members ahead of time, it cannot determine the correct solve order automatically.

Note: Dimensionally modeled relational (DMR) data sources query one or more underlying relational databases, and therefore cannot contain data source calculations. They are not considered dimensional data sources in the context of this topic.

Steps

1. Pause the pointer over the query explorer button and click the query that contains the data item for which you want to suppress values.
2. In the **Data Items** pane, click the data item.
3. In the **Properties** pane, in the **Calculation Intersection** property, click **Show "Not applicable"**.

When a query-defined calculated member and a data source calculated member intersect, the resulting value is suppressed. If the intersecting members do not contain data source calculations, the value is not suppressed.

Format Data

Format data in a report to improve readability. For example, you can show all date values in the order year, month, and day. If you do not set properties here, data is formatted according to the

properties set in the model. If the properties were not set in the model, data is formatted according to the International Components for Unicode (ICU) formats.

You can also format data based on a condition (p. 254) or specify the format for a particular object (p. 195).

When using a Cognos cube as a data source, mixed currency values use the asterisk character (*) as the unit of measure. Mixed currency values occur when you calculate values with different currencies.

The same behavior occurs for SAP BW data sources.

Steps

1. From the **Data** menu, click **Default Data Formats**.
2. In the **Format type** box, click a format type.

   The properties that you can set for the selected format type appear in the **Properties** box.
3. If you clicked the **Currency** format type and require different currencies in the report, click the add button . In the **Currencies** dialog box, select the currencies you want and click **OK**.

   For example, you may have one column whose values are in euros and another column whose values are in US dollars.
4. Set the properties that you want.

   If you added currencies in step 4, click each one and set the properties you want. If you did not add any currencies, any properties you set will apply to all currencies.

   For properties in which you type meta-characters that represent certain types of information, such as YYYY-MM-DD for dates, the meta-characters required depend on the authoring language specified for the report. For more information, see "Using Patterns to Format Data" (p. 641).

   If you specify a value for the **Pattern** property, all other formatting properties are ignored, with the following exceptions:

   - **Missing Value Characters**
   - **Zero Value Characters**
   - **Negative Pattern**

   Some properties are language-sensitive and should only be changed with caution.
5. Click **OK**.

The data formatting properties you set are applied to objects only in the current layout. If a data item contains values in multiple currencies, but only a subset of those currencies have defined formats, the default format for the locale you are working in is applied to values for which no format is specified.

Specify the Format for an Object

Specify the format for a particular object if you are not getting the results you want. For example, you add a measure to a report, and you want to see two decimals when you run the report. You set the number of decimals to two for the **Number** format type for the current layout. However, when you run the report, you see more than two decimals for the measure. To get the results you want, you must map the measure to the **Number** format type.

Steps

1. Click the object.
2. In the **Properties** pane, double-click the **Data Format** property.
3. Under **Format type**, click the format type you want to apply to the object.
4. To override any of the properties of the format type that were defined for the current layout, in the **Properties** box, click the property and specify its value.
5. Click **OK**.

Specifying the Number of Decimals

When specifying the number of decimals, Cognos 8 uses the IEEE 754 default rounding mode known as half even. With half even rounding, numbers are rounded toward the nearest truncated value, unless both truncated values are equidistant, in which case the value ending in an even digit is chosen, as follows:

- if the digit immediately after the precision to be shown is greater than 5, the number is rounded up.
- if the digit immediately after the precision to be shown is less than 5, the number is rounded down.
- if the digit immediately after the precision is a 5, the number is rounded down when the preceding digit is even and rounded up when the preceding digit is odd.

For example, if you specify no decimals, the number 78.5 is rounded to 78, while the number 73.5 is rounded to 74.

In addition, if the maximum number of decimals is lower than the actual number of decimals in the number, the number is rounded to the maximum number of decimals.

Locale-sensitive Properties

Report Studio contains an extensive library of customized properties adapted to users from different regions who speak different languages. If, for example, a modeler specifies that a given data item is a currency, only the proper currency symbol must be specified. When reports are created, Report Studio automatically adapts the format of the currency numbers to each user according to the content language specified in Cognos Connection.

When modelers or report authors specify properties, these properties override user preferences and risk creating inconsistent formatting for users of other cultures. It is usually much easier and safer to let Report Studio take care of formatting. For example, for the date format type, different regions use different characters to represent the date separator. If you specify a date separator, you may confuse users in other regions.

The following data formatting properties are locale-sensitive.

- **Currency Symbol Position**
- **Date Separator**
- **Date Ordering**
- **Calendar Type**
- **Time Separator**
- **Display AM/PM Symbols**
- **Clock**
- **Decimal Symbol**
- **Negative Sign Position**
- **Thousands Separator**
- **Group Size (digits)**
- **Secondary Group Size (digits)**
- **Missing Value Characters**
- **Zero Value Characters**
- **Pattern**
- **Negative Pattern**

Insert a Single Data Item

You can insert a single data item anywhere in your report using the singleton object. The singleton object retrieves only the first row value for that query. Inserting a single data item is useful when you want to show a value that is independent from the rest of the values in the report or when you want to insert some boilerplate text, such as a company name and address. For example, you can add the total revenue value in the header of each page in a report.

You can associate multiple singleton objects to a single query (p. 213) to optimize performance, such as when all the data items in the singleton are from the same database table. In addition, two or more singletons can reference data items from the same query. This is useful when using a single query is more efficient to display a set of single values than using multiple queries.

You can also filter the data item in the singleton. For example, you can show the total revenue for only the year 2007.

Queries that are associated to a singleton object are not supported when producing report output in delimited text (CSV) format.

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Singleton** to the report.

   An empty data container is created.

2. From the **Insertable Objects** pane, on the **Source** tab, drag a data item into the **Singleton** container.

   Tip: To create a singleton, you can also drag a data item directly to anywhere in your report layout.

3. If you want to change the query associated to the singleton object, in the **Properties** pane, double-click the **Query** property.

When the report is run, the first row value for the data item is retrieved.

Specify the List of Data Items for an Object

Specify the list of data items for an object when you must reference a data item that is in a query but is not in the layout. For example, you want to add a layout calculation to a list that uses a data item that is in the query in its definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work.

You must also specify the list of properties if you apply conditional formatting (p. 254) that uses a data item that is not in the query.

Steps

1. Click the layout object that you want.

   Tip: For a list of layout objects for which you can specify a list of properties, see the **Properties** property in "Report Studio Object and Property Reference" (p. 523).

2. In the **Properties** pane, double-click the **Properties** property.

3. Select the data items you want and click **OK**.

Show Data for a Specific Time Period

Show data for a specific time period to associate a report with a business time period rather than the execution time. For example, you have a monthly report that you run at the beginning of each month, and you want the last business day of the previous month to appear rather than the day on which you run the report.

If the report functions AsOfDate() and AsOfTime() are added to the report, they will return a value based on the results of the **As of Time Expression** object. If the **As of Time Expression** object is not added to the report, these two functions return the date and time at which the report is run.

If the **As of Time Expression** object is added more than once in the report, the first occurrence of the object in the layout that returns a valid value is used.

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **As of Time Expression** to the report.
2. Double-click **As of Time Expression**.
3. In the **Expression Definition** box, type the expression that you want and click **OK**.

   The expression must return a date-time value.

   Tip: You can drag a function that returns a constant from the **Constants** folder in the **Functions** tab. You can then change the constant to the value that you want to use.

   If you do not specify a time, the default time 12:00:00.000 AM is used.

Specify the Default Measure

You can specify the default measure for crosstabs and certain chart types. Specify the default measure to be used when the measures cannot be determined by what is on the edges. For example, you create a crosstab with Order method as rows and Product line as columns. You add Quantity and Revenue as nested rows, making Order method their parent. You then add Country under Order method. Since there is no measure specified for Country, you specify the default measure so that data is returned for each country.

In crosstabs, Report Studio automatically sets the default measure when you insert a measure into the crosstab cells.

Steps

1. To specify the default measure for a crosstab, do the following:
   - Click any part of the crosstab, and then click the select ancestor button in the title bar of the **Properties** pane.
   - Click **Crosstab** or the chart type.
   - Click the **Default Measure** property and click the measure you want to use as the default measure.
2. To specify the default measure for a chart, drag the measure that you want to the default measure box in the chart.

Specify What Appears for a Data Container that Contains No Data

You can specify what appears for a data container when there is no data available. For example, you can have text appear that states that there is no data available for a list object in a report. This can be done for lists, crosstabs, charts, maps, repeaters, repeater tables, and table of contents objects.

Steps

1. Click a data container.
2. In the **Properties** pane, click the select ancestor button and click the data container type.

   For example, if the data container is a list, click **List**.
3. Set the **No Data Contents** property to **Yes**.

   Two tabs appear at the top of the data container.
4. Click the no data contents tab .
5. In the **Insertable Objects** pane, on the **Toolbox** tab, insert the objects that you want to appear when there is no data available.

Adding Prompts

You can add prompts to a report to add interactivity for users. Prompts act as questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a product type. Only products belonging to the selected product type are retrieved and shown in the report.

Prompts are composed of three interrelated components. Parameters, which are based on parameterized filters, form the questions you want to ask users. Prompt controls provide the user interface in which the questions are asked. Finally, parameter values provide the answers to the questions.

Report Studio provides several ways to create prompts. You can

- define prompts using context filters
- use the Build Prompt Page tool
- build your own prompt and prompt page
- create a parameter to produce a prompt
- insert prompts directly into the report page

If you include reports from different packages in an interactive dashboard that uses global filters, ensure that you use the same parameter name for the prompt in all the reports. For more information about creating interactive dashboards, see the *Administration and Security Guide*.

You can also create prompts in the package. For more information, see the Framework Manager *User Guide*.

Define Prompts Using Context Filters

If context filters (or slicers) are defined, you can use these filters to create prompts in the report. Prompts provide questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a region. Only data for the specified region is retrieved and shown in the report.

Before you define a prompt, you must create context filters (p. 174).

Steps

1. In the **Context filter** section of the overview area, click the down arrow next to the filter that you want to use as a prompt.
2. Click **Prompt**.
3. Do one of the following:
   - To remove the prompt, click **No Prompt**.
   - To allow users who run the report to select any member from a tree of the hierarchy, click **Prompt on Hierarchy**.
   - To allow users who run the report to select from a list of members from the current level, click **Prompt on Level**.

Use the Build Prompt Page Tool

Use the **Build Prompt Page** tool to quickly add prompts to a report. Report Studio creates the prompts for you in a default prompt page.

Steps

1. Click the column you want users to be prompted on.

   To create multiple prompts or a cascading prompt, click more than one column.
2. From the **Tools** menu, click **Build Prompt Page**.

   A prompt page is created that has:
   - a page header
   - a prompt control for each selected column
   - a page footer containing **Cancel**, **Back**, **Next**, and **Finish** prompt buttons

You can add more objects or modify existing elements. For example, you can change the prompt control chosen by Report Studio.

Build Your Own Prompt and Prompt Page

Create your own prompt and prompt page to control how they appear in a report.

Steps

1. Pause the pointer over the page explorer button and click **Prompt Pages**.
2. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Page** to the **Prompt Pages** box.
3. Double-click the page you just created.
4. In the **Insertable Objects** pane, on the **Toolbox** tab , drag one of the following prompt controls to the prompt page.

Prompt control	Description
Text Box Prompt	Retrieves data based on a value that you type. Use this control when users know exactly what value they want to enter, such as a name or account number.
Value Prompt	Retrieves data based on values that you select from a list. Use this control to show the list of possible values from which users can choose. **Note:** The maximum number of items that can appear in a list is 5000.
Select & Search Prompt	Retrieves values based on search criteria that users specify. Data is then retrieved based on values users select from the search results. Use this control instead of a value prompt if the list of values is very long, which can slow down performance. **Tip:** Users have the option of performing a case sensitive or case insensitive search. A case sensitive search is faster, while a case insensitive search usually returns more values. You cannot use this control if you are working with SAP BW data sources.
Date Prompt	Retrieves data based on a date that you select. Use this control when you are filtering a date column. With this control, users can retrieve data for a specific day, a set of days, or a range of days.

Prompt control	Description
Time Prompt	Retrieves data based on a time that you select. Use this control to restrict a report to a particular time or time range. For example, you can use this control to see how many orders are received after business hours. You can then use this information to determine the number of staff needed to work after hours.
Date & Time Prompt	Retrieves data based on a date and time that you select. Use this control when you are filtering a datetime or timestamp column. This control is useful for specifying ranges. For example, you can retrieve all orders received from Monday at 12:00 a.m. to Friday at 5:00 p.m.
Interval Prompt	Retrieves data based on a time interval that you specify. Use this control to retrieve data that is related to the passage of time. For example, you can use this control to retrieve a list of products that were returned 30 or more days after they were purchased.
Tree Prompt	Retrieves data based on values you select from a list. Values are organized hierarchically. This control is useful when you are working with dimensional data sources. Data is shown from the top of a dimension hierarchy to the most detailed member, and users can choose the level of detail they want to view in the report. For more information about tree prompts, see "Control the Data That Appears in a Tree Prompt" (p. 212).
Generated Prompt	Selects a prompt control based on the data type of the data item. This control acts like a placeholder. When you run the report, the control is replaced by the appropriate prompt control. For example, if you are prompting date values, the control is replaced by a date & time prompt.

The **Prompt Wizard** dialog box appears.

5. If you are creating a text box, date, time, date and time, interval, or generated prompt, do the following:
   - Create a new parameter for the prompt or use an existing parameter.
   - Click **Next**.
   - If you create a new parameter, define the expression by selecting a data item from the package and the operator you want to use.

Tip: Make the prompt optional by selecting the **Make the filter optional** check box. When selected, users will not be required to choose a value when the report is run.

- Go to step 8.

6. If you are creating a value, select & search, or tree prompt, do the following:
   - Create a new parameter for the prompt or use an existing parameter.
   - Click **Next**.
   - If you created a new parameter, and you want to use the parameter to filter data, select the **Create a parameterized filter** check box and define the expression by selecting a data item from the package and the operator you want to use.

     If you are creating a tree prompt, you must choose **in** in the **Operator** box.

     You can use parameters for other purposes besides filtering data. For example, you can use a parameter to provide a value for a layout calculation, such as showing a user's name in the report. When the report is run, you can use a parameter to prompt the user to type his name and have it appear in the report.

     Tip: Make the prompt optional by selecting the **Make the filter optional** check box. When selected, users will not be required to choose a value when the report is run.
   - Click **Next**.
   - If you created a parameterized filter and you have more than one query defined in the report, select the query you want to filter and then click **Next**.
   - Select the **Create new query** check box to create the query that will be used to build the list of data values shown when the report is run.

     Tip: Do not create a new query if you want to link the prompt to an existing query or if you intend to create the query at a later time.
   - Click the ellipsis (...) button beside **Values to use** and click the data item you want to prompt on.

     Tip: You can choose a data item that is different than what users will see when they are prompted (see the following step). For example, you want to prompt on Product line. The package contains the data items Product line and Product line code. You can choose Product line code as the values to use for faster data retrieval.
   - Click the ellipsis (...) button beside **Values to display** and click the data item that you want users to see when they are prompted.
   - To create a cascading prompt, click the **Cascade Source** box, and select the parameter that represents the cascade source.
7. Click **Finish**.

The prompt control is added to the prompt page. A prompt page is like a report page. You can insert graphics, text, and so on, and apply any formatting you want.

You can also modify the properties of the prompt control by clicking it and making changes in the **Properties** pane.

Example - Create a Report Showing Products Shipped for a Specific Time Interval

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows the time interval between closing and shipping dates.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper-right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)** and **Sales (query)**.
6. Expand **Sales order** and add **Order number** to the list by double-clicking it.
7. Expand **Product** and add **Product name** to the list by double-clicking it.
8. Expand **Sales fact**. Add **Quantity** to the list by double-clicking it.
9. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Query Calculation** to the right of **Return quantity** in the list.
10. In the **Name** box, type the following and click **OK**:

    Time Interval
11. In the **Expression Definition** box, type the following and click **OK**:

    [Sales (query)].[Time dimension (close day)].[Date (close day)]-[Sales (query)].[Time (ship date)].[Date (ship date)]
12. From the **Data** menu, click **Filters** .
13. Click the add button.
14. In the **Expression Definition** box, type

    [Sales (query)].[Time dimension (close day)].[Date (close day)]-[Sales (query)].[Time (ship date)].[Date (ship date)] > ?p1?
15. Click **OK** twice.

    A parameterized filter is created that will return data when the difference between the return date and the order date is greater than the value specified by the user.
16. Pause the pointer over the page explorer button and click **Prompt Pages**.

17. In the **Insertable Objects** pane, drag **Page** to the **Prompt Pages** pane, and then double-click it.
18. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Interval Prompt** to the work area.

    The **Prompt Wizard** dialog box appears.
19. Click **Use existing parameter**, and then click **p1**.
20. Click **Finish**.
21. Run the report.

    An interval prompt appears.
22. In the **Days** box, type a value and click **Finish**.

    Tip: You can also type values for the **Hrs** and **Mins** boxes.

A list report appears, showing all products that were shipped after the time interval you specified. For example, if you type 5 days, the list will show products that were shipped more than 5 days after the order date.

Order number	Product name	Quantity	Date (ship date)	Time Interval
100001	Flicker Lantern	256	Jan 19, 2004	29 days
100005	Bear Survival Edge	120	Jan 19, 2004	9 days
100005	Double Edge	382	Jan 19, 2004	9 days
100005	Lady Hailstorm Titanium Woods Set	26	Jan 19, 2004	9 days
100005	Mountain Man Extreme	24	Jan 19, 2004	9 days
100005	Seeker Extreme	65	Jan 19, 2004	9 days
100011	BugShield Natural	1,792	Jan 19, 2004	10 days
100011	Compact Relief Kit	312	Jan 19, 2004	10 days
100011	Glacier Deluxe	98	Jan 19, 2004	10 days
100011	Hibernator Self - Inflating Mat	242	Jan 19, 2004	10 days
100011	Single Edge	2,490	Jan 19, 2004	10 days
100011	Star Peg	1,309	Jan 19, 2004	10 days
100011	Sun Shelter 15	1,164	Jan 19, 2004	10 days
100011	TrailChef Water Bag	2,172	Jan 19, 2004	10 days
100018	Blue Steel Max Putter	175	Jan 20, 2004	16 days
100018	Course Pro Gloves	439	Jan 20, 2004	16 days
100018	Course Pro Putter	411	Jan 20, 2004	16 days
100018	Hailstorm Titanium Woods Set	47	Jan 20, 2004	16 days
100021	Aloe Relief	189	Jan 19, 2004	14 days
100021	Blue Steel Putter	128	Jan 19, 2004	14 days

Create a Parameter to Produce a Prompt

Report Studio can automatically generate prompted reports based on parameters you create. When you run the report, Report Studio can generate a prompt page for each parameter not associated to an existing prompt page, depending on whether the prompt run option (p. 66) is selected or not.

You also define parameters when you want to create a drill-through report (p. 273), and you can use parameters to define master detail relationships (p. 282).

Steps

1. From the **Data** menu, click **Filters**.
2. On the **Detail Filters** tab, click the add button.

   The **Detail Filter** dialog box appears.

3. In the **Available Components** box, click the source or **Data Items** tab to select the data item you want to use for the prompt:
   - To filter data based on data items not shown in the report, double-click a data item on the **Source** tab .
   - To filter data that appears in the report but not necessarily in the model, such as calculations, double-click a data item on the **Data Items** tab .

   The data item appears in the **Expression Definition** box.
4. In the **Expression Definition** box, type an operator after the data item, or select an operator from the **Functions** tab.

   The operator sets some of the default properties of the prompt. For example, if the operator is equals (=), users will be able to select only a single prompt value, and the prompt's **Multi-Select** property is set to **No**.

   For more information about creating expressions, see "Using the Expression Editor" (p. 357).
5. Type a name after the operator to define the prompt parameter. A question mark must precede and follow the name.
6. Click **OK**.
7. In the **Usage** box, click one of the following:
   - To create a required prompt, click **Required**.
   - To create an optional prompt, click **Optional**.
   - To specify not to use the prompt, click **Disabled**.
8. Click **OK**.

Create a Prompt Directly in a Report Page

You can add prompt controls directly in a report page instead of creating a prompt page.

Prompt controls that are added to report pages will not appear in the following:

- saved reports
- PDF reports
- reports that are sent to users by email
- scheduled reports

Prompt controls are by their very nature interactive. They are used to satisfy parameter values before running a report. As a result, prompt controls added to a report page only appear when you run the report in HTML format. When you run a report in HTML format, users select which values they want to see, and the report is refreshed, producing a new report.

For the non-interactive reports listed above, prompt parameter values must be collected and satisfied before the report is run. You provide the parameter values you want using the **Run options** tab in Cognos Connection. If you do not provide all the required values, the report will fail to run. You can access the Run options tab by clicking the set properties button for the report.

If you are using the sample reports that come with Report Studio, the reports Consumer Trends, GO Media, Product Line by Year-prompt, and Sales Representative Contact List contain prompt controls on report pages.

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag a prompt control to the location where you want it to appear.

   The **Prompt Wizard** dialog box appears.

2. Provide the information necessary to create the prompt.

   Tip: Make the prompt optional by selecting the **Make the filter optional** check box. Otherwise, when you run the report for the first time, the prompt appears in a generated prompt page rather than in the report page. Alternatively, you can specify a default selection for the prompt (p. 210).

3. In the work area, click the prompt.

4. In the **Properties** pane, set the **Auto-Submit** property to **Yes**.

   If you do not set the **Auto-Submit** property to **Yes**, the report will not refresh when users select a different prompt value.

   Tip: An alternative to setting the prompt to auto-submit is to add a prompt button from the **Toolbox** tab and setting its **Type** property to **Finish**.

The prompt control is added to the report page. You can modify its properties by clicking it and making changes in the **Properties** pane.

Modifying Prompts

For each prompt you create, you can modify its properties by specifying values in the **Properties** pane. For example, you can

- change the prompt control interface
- specify that a prompt requires user input
- allow users to select multiple values
- show or hide prompt status
- specify default selections
- specify prompt values
- add a prompt button

- create a cascading prompt

Some properties you set for a prompt may be overridden under some conditions. For example, some properties set for the filter associated with a prompt may override the corresponding prompt property.

Change the Prompt Control Interface

By default, when you create a prompt, Report Studio selects the prompt control interface. You can change the prompt control interface to something different, depending on the type of prompt you created. For example, for a value prompt, you can choose a drop-down list, a list box, or a radio button group.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Select UI** property and click the interface you want.

Specify That a Prompt Requires User Input

You can specify that a prompt requires user input before the report can run.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Required** property and click **Yes**.
3. Pause the pointer over the page explorer button and go to a report page.
4. From the **Data** menu, click **Filters** .
5. Click the filter associated with the prompt.
6. In the **Usage** box, click **Required** and then click **OK**.

When you run the report, a star appears next to the prompt indicating that the user must select or type a value.

If you have a cascading prompt and the parent prompt control is required, the child prompt control is disabled. This ensures that users choose at least one value in the parent prompt before they can choose a value in the child prompt. Conversely, if the parent control is optional, then the child control is populated. This gives users the ability to choose values in the child prompt without having to choose a value in the parent prompt.

Allow Users to Select Multiple Values in a Prompt

You can allow users to select more than one value or a range of values in a prompt. For example, you have a prompt for which users must select a product line. You can modify the prompt so that users can select more than one product line.

If you enable multiple selections, the **Auto-Submit** property is always set to no.

Steps

1. Click the prompt control that you want to modify.
2. In the **Properties** pane, choose whether to allow users to specify more than one value or a range of values:
   - To allow users to specify more than one value, set the **Multi-Select** property to **Yes**.
   - To allow users to specify a range of values, set the **Range** property to **Yes**.
3. Pause the point on the page explorer button and go to a report page.
4. From the **Data** menu, click **Filters** .

   If you have more than one query defined in the report, you must first click an object linked to a query. If there is more than one query defined in the report, click an object that is linked to the query you want.
5. Double-click the filter associated with the prompt.
6. Change the operator to one of the following:
   - If you are creating a multi-select prompt, change the operator to `in`.

     For example,[Product_line] in ?Product line?, where [Product_Line] is the name of the data item, allows users to select multiple product lines.
   - If you are creating a range prompt, change the operator to `in_range`.

     For example, [Margin] in_range ?Margin?, where [Margin] is the name of the data item, allows users to specify a margin range.
7. Click **OK** twice.

Show or Hide Prompt Status

Each prompt you create in a report provides dynamic validation when the report is run. Validity checks are performed to ensure that the data is correct, and that required values are supplied. For example, a star appears next to each required prompt. An arrow appears next to a prompt if you must select or type a value. If you type an incorrect value, a dotted line appears. You can choose whether to show the star and arrow for each prompt.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Hide Adornments** property and click **Yes** to hide the prompt characters or **No** to show them.

Specify a Default Selection for a Prompt

You can specify a default selection for a prompt so that users do not have to select or type a value when they run the report.

Steps

1. Click the prompt control that you want to modify.
2. If you want to define a range of values, in the **Properties** pane, set the **Range** property to **Yes**.
3. If you want to specify more than one default selection, set the **Multi-Select** property to **Yes**.
4. In the **Properties** pane, double-click the **Default Selections** property.
5. Click the add button , and do one of the following:
   - If you chose to define a single value, type the value you want as the default selection.
   - If you chose to define a range of values, type the minimum and maximum values of the range in the **Minimum value** and **Maximum value** boxes, respectively.
6. Click **OK**.
7. Repeat steps 4 to 5 to specify other default selections.
8. Click **OK**.

Specify Prompt Values

Provide your own values in a prompt

- to show something different from what is in the database
- to improve performance by not accessing the database
- to provide text for optional prompts, such as Select a value
- to restrict the number of values available

For example, you have a prompt in which users choose a country. For the database value United States, you want USA to appear in the prompt.

Steps

1. Click the prompt control that you want to modify.
2. In the **Properties** pane, double-click the **Static Choices** property.
3. Click the add button .
4. In the **Use** box, type the prompt value you want to add.
5. In the **Display** box, type the value that will appear in the prompt.
6. Click **OK**.

7. Repeat steps 4 to 7 to add other prompt values.
8. To link a prompt value to a condition (p. 254), do the following:
   - Click the **Variable** drop-down list and choose the variable you want to use or create your own.

     For information about creating variables, see "Add a Variable" (p. 259).
   - Click the **Value** drop-down list and click one of the possible values for the variable.
   - Click the static value you want to link to the variable and click the edit button .
   - In the **Display** box, type the value that you want to appear in the prompt for that particular value of the variable.
   - Repeat this procedure for each additional value.
9. Click **OK**.

Add a Prompt Button

Add prompt buttons so that users can submit selected items, cancel a report, or navigate between pages.

When you are building prompts and prompt pages (p. 201), you may have to add a prompt button to submit selections. Some prompt controls, such as the value prompt, can be set to submit selections automatically. Other prompt controls, such as the date prompt, require a prompt button.

Steps

1. Pause the pointer over the page explorer button and click the report or prompt page that contains the prompt control for which you want to add a prompt button.
2. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Prompt Button** to the work area.
3. Click the prompt button and in the **Properties** pane, click the **Type** property. Choose one of the following actions.

Goal	Action
Cancel the report	**Cancel**
Go to the previous prompt page	**Back**
Go to the next prompt page	**Next**
Run the report	**Finish**

Goal	Action
Reprompt the user **Tip:** Reprompting users is useful when you have cascading prompts.	**Reprompt**

Create a Cascading Prompt

Create a cascading prompt to use values from one prompt to filter values in another prompt. For example, a report contains the columns Product line and Product type. You create prompts for these columns, and you specify that the Product type prompt is a cascading prompt that uses Product line as the cascading source. When users select a product line, they see only the product types related to the selected product line.

Steps

1. To make the cascading source a required prompt, click it and in the **Properties** pane, set the **Required** property to **Yes**.
2. Click the prompt control that you want to use as a cascading prompt.
3. In the **Properties** pane, double-click the **Cascade Source** property.
4. Click the parameter that represents the cascade source, and click **OK**.
5. If the prompt allows users to select multiple values, add a prompt button to the cascade source to provide the cascading prompt with the appropriate values:
   - In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Prompt Button** to the report.
   - Click the prompt button and in the **Properties** pane, click the **Type** property and click **Reprompt**.
   - To change the text in the prompt button, in the **Insertable Objects** pane, on the **Toolbox** tab, drag **Text Item** to the prompt button and type the text you want.

For more information about how to create cascading prompts, see the Report Studio Tour.

Control the Data That Appears in a Tree Prompt

You can control what data appears in a tree prompt and how the data is structured to get the results that you want. To do this, you add various functions to the filter expression.

In addition, the operator that you chose in the **Prompt Wizard** dialog box (p. 201) controls what appears next to each prompt value. If the operator is in or not in, check boxes appear next to each prompt value. If the operator is equals (=), no check boxes appear.

Steps

1. Pause the pointer over the query explorer button and click the query that is associated with the prompt.
2. In the **Data Items** pane, double-click the data item that you are prompting on.
3. In the **Expression Definition** box, type one of the following functions and click **OK**.

Goal	Function
Show the hierarchical structure of all members in the hierarchy. If this function is not used, all members are shown in a flat list.	rootmembers(*data_item*)
Show the descendants of the data item in a hierarchical structure, where x represents the level. For example, if you are prompting on the Year hierarchy, and x=1, you will see 2004, 2005, and 2006 in the tree. If x=2, you will see 2004 Q1, 2004 Q2, and so on.	descendants(rootmembers (*data_item*, x)
Show the children of a member. For example, 2004 Q1, 2004 Q2, 2004 Q3, and 2004 Q4 appears for the member 2004.	children(*member*)

Working with Queries

Queries specify what data appears in the report. In Report Studio, you create and modify queries using Query Explorer. Query Explorer provides an alternative way to modify existing reports or author new reports. You can use Query Explorer to perform complex tasks and other tasks that are more difficult to do in the report layout. For example, use Query Explorer to

- improve performance by changing the order in which items are queried from the database, or by changing query properties to allow the report server to execute queries concurrently where possible

  Note: By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the *Administration and Security Guide*.
- view or add filters and parameters, and modify their properties
- view or add dimensions, levels, and facts
- incorporate SQL statements that come from other reports or reports that you write

- create complex queries using set operations and joins

Relating Queries to Layouts

Queries and layouts work together. After you decide the type of data that you need, you must create a layout in which to show the results. Each column of data must be both selected for the query, and shown in a layout unless there are some columns that you don't want to show. The query and layout portions of a report must be linked to produce a valid report.

Report Studio automatically links query and layout. For example, when you use Report Studio and the list report template, query and layout are automatically linked. In addition, you can create a query and manually link it to a data container. In the layout, when you drag a data container to the work area, you can specify which query to use by clicking the **Query** property in the **Properties** pane. You then add the items in the query that you want to see in the layout by dragging them from the **Data Items** tab in the **Insertable Objects** pane.

Working with Dimensional Data Sources

When you are working with SAP BW data sources, you can use only a single hierarchy in a query.

Creating queries using a mix of OLAP and relational data is not supported. If you create queries using a database for which you do not know the type, consult your database administrator or modeler.

When performing multi-cube queries using dimensional data sources, the following restrictions apply:

- Only basic operators (+, *, /,-) are available for cross-cube calculations.
- Inner joins are not supported.
- All joins for multi-cube queries are outer joins.
- You cannot sort or filter on a conformed dimension (query subject).
- Conformed dimensions are created in Framework Manager.
- Viewing MDX using the **Generated SQL/MDX** query property may not show you the actual MDX that is executed. This is because when a multi-cube query is executed, a number of smaller queries are actually executed.

Connecting Queries Between Different Data Sources

Cognos 8 supports the following:

- RDBMS to RDBMS joins
- set operations of any two queries
- master detail relationships of any two queries
- drill from any query to any other query

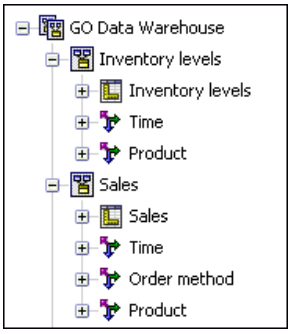
You cannot create the following types of joins:

- cube-to-cube (homogeneous)
- cube-to-cube (heterogeneous)
- cube-to-RDBMS
- cube-to-SAP BW
- SAP-BW-to-RDBMS

Multiple-fact Queries

You must understand the result of a query on more than one fact table to achieve the results that you want. The result of a multiple-fact query varies depending on whether you are working with conformed or non-conformed dimensions, as well as on the level of granularity and on the additive nature of the data.

The following data source has characteristics that affect results if you use a multiple-fact query with Inventory levels and Sales. Granularity for time differs in that inventory levels are recorded monthly and sales are recorded daily. Also, Sales includes a non-conformed dimension, Order method.



The following examples will help you interpret the results of a multiple-fact query and understand the options available for changing a query to obtain the results that you want.

Conformed Dimensions

Individual queries on Inventory levels and Sales by Quarter and Product yield the following results.

Quarter	Product	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	720	727	608
	Aloe Relief	1,128	1,236	1,256
	Course Pro Umbrella	1,077	1,217	902
200501		**975**	**1,060**	**2,766**
200502	Husky Rope 50	667	632	1,246
	Aloe Relief	988	874	1,710
	Course Pro Umbrella	969	1,025	2,158
200502		**874**	**843**	**5,114**

A query on multiple facts and conformed dimensions respects the cardinality between each fact table and its dimensions and returns all the rows from each fact table. The fact tables are matched on their common keys, Time and Product.

Product and Time apply to both Inventory levels and Sales. However, inventory levels are recorded monthly and sales are recorded daily. In this example, results are automatically aggregated to the lowest common level of granularity. Quantity, which comes from Sales, is rolled up to months.

Nulls are often returned for this type of query because a combination of dimensional elements in one fact table may not exist in the other. For example, if Husky Rope 50 was available in inventory in 200501, but there were no sales of this product in the same time period, Quantity would show a null in the Husky Rope 50 row.

Non-conformed Dimensions

If a non-conformed dimension is added to the query, the nature of the results returned by the query is changed.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	Telephone	720	727	254
	Husky Rope 50	Mail	720	727	70
	Husky Rope 50	E-mail	720	727	50
	Husky Rope 50	Web	720	727	98
	Husky Rope 50	Sales visit	720	727	136
	Aloe Relief	Telephone	1,128	1,236	400
	Aloe Relief	E-mail	1,128	1,236	138
	Aloe Relief	Web	1,128	1,236	378
	Aloe Relief	Sales visit	1,128	1,236	340
	Course Pro Umbrella	Fax	1,077	1,217	72
	Course Pro Umbrella	Telephone	1,077	1,217	140
	Course Pro Umbrella	E-mail	1,077	1,217	202
	Course Pro Umbrella	Web	1,077	1,217	80
	Course Pro Umbrella	Sales visit	1,077	1,217	224
	Course Pro Umbrella	Special	1,077	1,217	184
200501			**975**	**1,060**	**2,766**

Order Method exists only in Sales. Therefore, it is no longer possible to aggregate records to a lowest common level of granularity because one side of the query has dimensionality that is not common to the other side of the query. Opening inventory and Closing inventory results are repeated because it is no longer possible to relate a single value from these columns to one value from Quantity.

Grouping on the Quarter key demonstrates that the result in this example is based on the same data set as the query on conformed dimensions. Summary values are the same. For example, the total quantity for 200501 is 2,766 in both examples.

Filters on Non-conformed Dimensions

By default, to improve performance, filters are applied at the database level. The default behavior can result in unexpected nulls when you add a filter to a multiple-fact query. If you create a filter on the Order method dimension to show only the fax order method, and apply the filter at the data source, the report includes nulls.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50		720	727	
	Aloe Relief		1,128	1,236	
	Course Pro Umbrella	Fax	1,077	1,217	72
200501			**975**	**1,060**	**72**
200502	Husky Rope 50	Fax	667	632	82
	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
200502			**874**	**843**	**490**
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			**1,120**	**1,246**	**302**
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
	Course Pro Umbrella		1,520	1,388	
200602			**1,120**	**1,042**	**122**
Summary			**1,022**	**1,048**	**986**

The filter is applied only to one subject area, Sales. Because Order method does not exist in Inventory levels, all products still appear in the report. For example, Course Pro Umbrella was in inventory in 200602. Because there were no sales using the fax order method for this product in 200602, Order method and Quantity are null.

To remove the nulls, change the filter so that it is applied to the result set instead of the data source. By applying the filter to the result set, you get the same results that you would get if you were working with conformed dimensions. Only the products that were ordered using the fax order method appear in the report.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Course Pro Umbrella	Fax	1,077	1,217	72
200501			**1,077**	**1,217**	**72**
200502	Husky Rope 50	Fax	667	632	82
	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
200502			**874**	**843**	**490**
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			**1,120**	**1,246**	**302**
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
200602			**919**	**869**	**122**
Summary			**989**	**1,025**	**986**

The summary for quantity is 986 using either filter method, which shows the results are based on the same data set.

In Report Studio, there are two types of filters. A detail filter is applied to the data source. A summary filter is applied to the result set. For more information about how to apply detail and summary filters, see "Filter Data" (p. 172).

Add Queries to a Report

You can create multiple queries in Query Explorer to suit your particular needs. For example, you can create a separate query for each data container in a report to show different data.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.

2. In the **Insertable Objects** pane, drag one of the following objects to the work area.

Object	Description
Query	Adds a query.
Join	Adds a join relationship.
Union	Adds a union operator.
Intersect	Adds an intersect operator.
Except	Adds an except (minus) operator.
SQL	Adds SQL.
MDX	Adds MDX.

3. In the **Properties** pane, specify the properties you want.

   For example, if you added a join, double-click the **Join Relationships** property to define the join.

4. Double-click a query.

5. In the **Insertable Objects** pane, on the **Source** tab , drag data items that you want in the query to the **Data Items** pane.

   Tip: You can add data items to the query that you do not want to appear in the layout. For example, if you want to filter on Product line code but you want users to see Product line in the layout, you must add both data items to the query.

6. To create a new data item in the query, in the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Item** pane.

7. To add a filter, in the **Insertable Objects** pane, on the **Toolbox** tab, drag **Filter** to the **Detail Filters** pane or to the **Summary Filters** pane, and define the filter expression (p. 172).

   Tip: You can also create a filter by dragging a data item from the **Source** tab to the one of the filters panes and then completing the filter expression. If you are working with a dimensional data source, you can quickly filter data by dragging a member instead of the level to which the member belongs. For example, dragging the member 2006 from the Years level will filter data for the year 2006. This method is quicker than dragging the Years level and then completing the filter expression to specify to show data only for the year 2006.

When adding queries to the report

- right-click the work area and click **Show Package Sources** to see the queries that use data items from a package

- right-click the work area and click **Expand References** to see the relationships that exist between queries in the report, which is useful when you are creating complex queries

Create a Union Query

Create a union query to combine two or more queries into one result set.

You can combine queries that use different data sources. For example, you can combine a query that returns data from a dimensional data source with a query that returns data from a relational data source.

To combine two queries, the following conditions must be met:

- The two queries must have the same number of data items.
- The data types of the data items must be compatible and the data items must appear in the same order.

  For numeric data types, integer, float, double, and decimal are compatible.

  For string data types, char, varChar, and longVarChar are compatible.

  For binary data types, binary and varBinary are compatible.

  Date data types must match perfectly.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. In the **Insertable Objects** pane, do the following:
   - Drag **Query** to the work area.
   - Drag **Union**, **Intersect**, or **Except** to the right of the query.

     Two drop zones appear to the right of the operator.
   - Drag a **Query** object to each drop zone.

   Two queries are created in the work area and a shortcut to each query appears in the drop zones.
3. Double-click each query that makes up the union query and add the data items that you want.
4. On the toolbar, press the back button to return to the Queries work area.
5. Click the set operator that you added in step 3.
6. In the **Properties** pane, click the **Duplicates** property and choose whether you want to remove or preserve duplicate rows.
7. Double-click the **Projection List** property.

   The projection list shows the list of projected data items for the set operation.

8. If you want Report Studio to automatically produce the list of projected data items, click **Automatically generated**.

   Report Studio generates the projection list using only one of the two queries being unioned.

9. To add, delete, move, or rename data items in the projection list, click **Manual** and make the changes you want.

10. Click **OK**.

11. Double-click the union query.

12. In the **Insertable Objects** pane, on the **Source** tab, drag the data items that you want to add to the query to the **Data Items** pane.

The union query is complete. You can now link it to a data container in the layout.

Example - Create a Two-column List Report for Three Data Items

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a list report that shows revenue for all product lines and order methods. However, you want the product lines and order methods to appear in a single column. To create this report, you use a union query to join the Product line and Order method data items.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.

2. Click the **GO Data Warehouse (query)** link; click the **Launch** link in the upper-right corner of the page, and then click **Report Studio**.

3. In the **Welcome** dialog box, click **Create a new report or template**.

4. In the **New** dialog box, click **Blank** and click **OK**.

5. Pause the pointer over the query explorer button and click **Queries**.

6. In the **Insertable Objects** pane, do the following:

   - Drag **Query** to the work area.
   - Drag **Union** to the right of the query.

     Two drop zones appear to the right of the operator.
   - Drag a **Query** object to each drop zone.

     Query2 and **Query3** are created in the work area, and a shortcut to each query appears in the drop zones.

7. Double-click **Query2**.

8. In the **Insertable Objects** pane, on the **Source** tab, drag the following data items to the **Data Items** pane:

- from the **Product** folder, **Product line**
- from the **Sales fact** folder, **Revenue**

9. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Data Item** to the **Data Items** pane.

   The data item will be used to sort product lines and order methods in the report.

10. In the **Expression Definition** box, type **'A'** and click **OK**.
11. In the **Properties** pane, click the **Name** property, delete the default name, and type

    Sort key

12. On the toolbar, press the back button to return to the Queries work area.
13. Double-click **Query3**.
14. In the **Insertable Objects** pane, on the **Source** tab, drag the following data items to the **Data Items** pane:
    - **Order method**
    - **Revenue**

    Tip: You can find these items in the **Order method** and **Sales fact** folder.

15. Repeat steps 9 to 11 to create a Sort key data item in Query3 with 'B' as its definition.
16. On the toolbar, press the back button to return to the Queries work area.
17. Click the **Union** operator.
18. Double-click the **Projection List** property.

    The **Product line** item in the projection list contains both product lines and order methods.

19. Click **Manual**.
20. Click **Product line** and then click the edit button .
21. In the **Edit** box, type the following after **Product line**:

    & Order method

22. Click **OK** twice.
23. Double-click **Query1**.
24. In the **Insertable Objects** pane, on the **Source** tab, drag the following data items to the **Data Items** pane:
    - **Product line & Order method**
    - **Revenue**
    - **Sort key**

25. With **Sort key** selected, in the **Properties** pane, click the **Pre-Sort** property and click **Sort ascending**.

    When you run the report, all product lines will appear first, followed by all order methods.

26. Pause the pointer over the page explorer button and click **Page1**.

27. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **List** to the work area.

28. Click the list.

29. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.

30. Click the **Query** property and click **Query1**.

    The list is linked to the union query.

31. Double-click the **Properties** property.

32. Select the **Sort key** check box and click **OK**.

    Since the Sort key data item does not appear in the list, you must make it a property of the list before it can sort product lines and order methods.

33. In the **Insertable Objects** pane, on the **Data Items** tab, drag the following items from Query1 to the list:

    - **Product line & Order method**
    - **Revenue**

34. Run the report.

A list report with two columns is produced. All product lines and order methods appear in the first column.

Product line: & Order Method	Revenue
Camping Equipment	1,589,036,664.03
Golf Equipment	726,411,367.89
Mountaineering Equipment	409,660,132.9
Outdoor Protection	75,994,296.25
Personal Accessories	1,885,673,307.78
E-mail	179,843,044.16
Fax	70,073,542.01
Mail	46,091,338.97
Sales visit	310,194,834
Special	27,351,320.25
Telephone	340,985,781.06
Web	3,712,235,908.4

Create a Join Relationship

You can create a join relationship to join two queries.

In general, join relationships should be created in the Framework Manager model. Create a join relationship in Report Studio if what you are trying to do cannot be modeled in Framework Manager.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. In the **Insertable Objects** pane, do the following:
   - Drag **Query** to the work area.
   - Drag **Join** to the right of the query.

     Two drop zones appear to the right of the operator.
   - Drag a **Query** object to each drop zone.

     Two queries are created in the work area, and a shortcut to each query appears in the drop zones.
3. Double-click each query that makes up the join query and add the data items that you want.
4. On the toolbar, press the back button to return to the Queries work area.
5. Click **Join**.
6. In the **Properties** pane, double-click the **Join Relationships** property.
7. Click **New Link**.
8. Click a data item in the left query, and then click a data item in the right query to create a link.
9. For each query, click **Cardinality**, and then click the cardinality that you want to use.
10. Click **Operator**, and then click the operator that you want to use.
11. To convert the join relationship to an expression, click the **Convert to expression** button.

    Convert the join relationship to an expression to make changes to the join definition.

    Note: After you convert the relationship to an expression, you cannot change it back to a relationship.
12. Click **OK**.
13. Double-click the join query.
14. In the **Insertable Objects** pane, on the **Source** tab, drag the data items that you want to add to the **Data Items** pane.

The join query is complete. You can now link it to a data container in the layout.

Add Dimension Information to a Query

Add dimension information to a query to obtain the results that you want. Add dimension information if any of the following apply:

- There is no dimension information available in the data source.

  For example, the data source contains flat data that you want to model dimensionally.

- You want to override the dimension information in the data source.

- You want to extend or restrict dimension information in the data source.

If you do not add dimension information and the data source contains dimensions, then the dimension structure of the data source is used. If the data source contains no dimensions, Cognos 8 creates a default dimension structure.

Dimension information is not intended to define the presentation of information, but to help query planning. It can be considered as a form of query hint. In addition, users will not be able to drill down (p. 279) on dimensions that you add to a report.

Steps

1. Pause the pointer over the query explorer button and click the query you want.
2. In the **Properties** pane, click the **Override Dimension Info** property and click **Yes**.

   The **Dimension Info** tab appears in the work area.
3. Click the **Dimension Info** tab.
4. To create a dimension from an existing data item, in the **Insertable Objects** pane, on the **Source** tab , drag the data item to the **Dimensions** pane.

   Report Studio automatically generates the entire dimension.
5. To create a new dimension, in the **Insertable Objects** pane, on the **Toolbox** tab , drag **Dimension** to the **Dimensions** pane, and then build it by adding the objects you want:
   - To create a level, drag **Level**.
   - To create a level hierarchy, drag **Level Hierarchy**.
   - To create a member hierarchy, drag **Member Hierarchy**.

   For information about these objects, see "Working with Dimensional Data" (p. 167).
6. In **Insertable Objects** pane, on the **Source** tab, drag the data items you want to the objects you added in the previous step.

   For example, if you created a level, define a key by dragging a data item to the **Keys** folder.
7. To create a fact, in the **Insertable Objects** pane, on the **Source** tab, drag the data item you want to the **Facts** pane.

Example - Create a Crosstab Report that Shows Empty Rows

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a crosstab report that shows the total revenue per year for each product line, broken down by order method. By default, if no revenue was produced for a particular product

line in a specific year, no row appears in the crosstab for that product line and year. You override the dimension information of the crosstab so that empty rows appear in the report.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper-right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , drag the following data items to the crosstab:
   - **Product line** as rows
   - **Month** as nested rows
   - **Order method** as columns
   - **Revenue** as the measure

   Tip: You can find these items in the **Product**, **Time dimension**, **Order method** and **Sales fact** folders.
6. Right-click the crosstab and click **Go to Query**.
7. In the **Properties** pane, click the **Override Dimension Info** property and click **Yes**.

   The **Dimension Info** tab appears at the bottom of the work area.
8. Click the **Dimension Info** tab.
9. In the **Insertable Objects** pane, on the **Source** tab, drag the following items to the **Dimensions** pane:
   - **Product line**
   - **Date**

   Product line and **Date** become separate dimensions in the query.
10. Pause the pointer over the page explorer button and click **Page1**.
11. Click **Month**.
12. In the **Properties** pane, double-click the **Sort** property.
13. In the **Data Items** box, drag **Date** to the **Sort List** box and click **OK**.
14. Run the report.

All order years appear for all product lines, even if no revenue was produced.

Revenue		Fax	Sales visit	Mail	E-mail	Web	Special	Telephone
Personal Accessories	April	1,964,196.52	4,689,359.81	629,825.45	4,042,713.02	141,264,184.06	328,897.8	7,447,517.44
	August	1,249,763.59	3,128,830.46	307,113.77	2,832,006.11	104,332,696.92	739,717.33	5,319,924.04
	December	1,145,604.11	3,211,826.32	302,749.13	1,917,882.1	128,221,947.72	430,771.39	4,236,870.14
	February	1,806,737.02	4,163,054.2	568,455.22	4,429,247.89	152,432,562.34	481,876.27	6,647,017.67
	January	895,034.77	5,474,576.56	601,395.48	4,711,659.65	135,431,248.84	753,586.22	7,523,100.08
	July	1,762,645.35	4,734,887.51	717,363.93	3,369,818.96	160,106,945.11	241,531.1	5,827,903.95
	June	2,168,907.29	4,357,653.94	857,114.52	4,547,280.16	189,657,211.81	202,151.21	6,335,525.02
	March	2,090,675.43	4,345,899.75	647,327.76	4,741,322.86	173,389,658.56	384,809.31	9,604,518.24
	May	1,539,129.6	4,901,413.4	500,489.34	4,285,643.1	165,597,071.61	505,986.42	7,780,551.57
	November	865,655.49	2,625,708.74	430,179.08	2,699,549.43	114,924,163.19	435,513.59	4,636,929.37
	October	1,268,073.18	2,921,343.75	344,630.15	2,576,794.22	109,013,588.75	271,584.59	4,015,217.08
	September	1,206,563.11	3,140,888.01	512,713.2	2,497,169.04	117,864,894.53	410,203.27	4,146,559.76
Camping Equipment	April	2,144,229.31	16,438,531.47	1,691,221.19	7,237,332.94	113,672,629.42	960,469.42	13,471,127.07
	August	1,569,930.02	10,916,363.73	1,890,970.77	5,561,984.75	75,773,253.58	629,564.93	14,079,719.36
	December	989,602.41	10,848,536.96	1,228,357.23	4,634,727.99	78,382,512.19	1,242,770.05	10,304,096.72
	February	1,746,949.32	19,015,437.03	2,869,223.69	7,908,320.81	104,431,827.5	1,255,422.44	16,474,364.69
	January	2,911,793.84	17,400,183.06	2,220,230.79	8,819,237.89	99,254,570.4	1,477,428.51	14,640,710.7
	July	2,107,481.64	18,482,223.22	1,699,617.67	6,703,357.87	118,240,121.54	938,502.72	13,252,194.39
	June	2,254,995.04	17,728,480.87	2,283,339.25	7,572,966.77	112,352,348.8	1,203,878.29	14,906,200.45
	March	2,197,710.1	15,455,138.41	1,788,317.75	5,895,621.88	104,589,002.39	1,913,000.46	12,316,369.47

Reference Package Items in Child Queries

When you create a child query in Report Studio, you can only reference items from its parent or from other queries. For example, if you add a filter to a child query, the only items that you can insert into the expression are items that exist in other queries defined in the report. To add an item from the package, you must unlink the child query from its parent.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. Unlink the child query from its parent by selecting the parent query shortcut to the right of the child query and clicking the delete button.
3. Double-click the child query.

   Package items are now available on the **Source** tab of the **Insertable Objects** pane.
4. Add the package items you want.

   For example, to create a detail filter that references a package item, drag the filter object from the **Toolbox** tab to the **Detail Filters** pane, and then add the items you want in the **Expression Definition** box.
5. Click the up arrow in the toolbar to return to the queries work area.
6. Recreate the link between the child query and the parent query by dragging the parent query to the right of the child query.
7. If necessary, double-click the child query to complete it.

   For example, if you created a filter, you may need to complete the filter expression.

Working with SQL or MDX

For each query in a report, you can work with the SQL or MDX that is executed when you run a report. You can

- view the SQL or MDX
- edit the SQL or MDX
- build a report using your own SQL or MDX
- choose the data source

When working with a relational data source, the SQL that is produced by Report Studio depends on the report output selected.

View the SQL or MDX

View the SQL or MDX to see what is passed to the database when you run a report.

Steps

1. To view the SQL or MDX for the entire report, from the **Tools** menu, click **Show Generated SQL/MDX**.

   This option shows the SQL or MDX that will be executed in the data source. The SQL or MDX is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.

2. To view the SQL or MDX for a specific query, do the following:

   - Pause the pointer over the query explorer button and click the query you want.
   - In the **Properties** pane, double-click the **Generated SQL/MDX** property.

   The **Generated SQL/MDX** property shows the SQL or MDX that is executed when you view tabular data (**Run** menu, **View Tabular Data**), which shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.

The SQL or MDX for the query appears in the **Generated SQL/MDX** dialog box. For SQL, you can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos SQL, which is a generic form of SQL that Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Edit the SQL or MDX

You can edit the SQL or MDX for a query to suit your particular needs. For example, you may want to edit the SQL or MDX to improve performance or to troubleshoot problems.

If you did not originally build the query using SQL or MDX, you must convert the query to SQL or MDX before you can make changes. Converting a query to SQL or MDX is an irreversible process.

Steps

1. Pause the pointer over the query explorer button and click the query you want.
2. In the **Properties** pane, double-click the **Generated SQL/MDX** property.
3. Click **Convert**.
4. Make the changes you want.

   If you are working with MDX, you must be aware of the MDX syntax that Report Studio supports.
5. Click **Validate** to check for errors.
6. Click **OK**.

Build a Report Using Your Own SQL or MDX

You can build a report by adding SQL or MDX from an external source, such as another report.

If you are working with MDX, you must be aware of the MDX syntax that Report Studio supports.

Steps

1. From the **File** menu, click **New**.
2. Click **Blank** and click **OK**.
3. Pause the pointer over the query explorer button and click **Queries**.
4. In the **Insertable Objects** pane, drag **Query** to the work area.
5. In the **Insertable Objects** pane, do one of the following:
   - To build a SQL query, drag **SQL** to the right of the query.
   - To build an MDX query, drag **MDX** to the right of the query.

   Tip: You can drag **SQL** or **MDX** anywhere in the work area and Report Studio will automatically create a query.
6. In the **Properties** pane, double-click the **Data Source** property to choose the data source.
7. Click the data source you want and click **OK**.
8. If required, in the **Catalog** property, type the name of the catalog that you want to use.
9. Double-click the **SQL** or **MDX** property.
10. Type the SQL or MDX you want to use.

11. Click **Validate** to check for errors.
12. Click **OK**.
13. Double-click the query.

    If the SQL or MDX is valid, data items defined in the SQL or MDX appear in the **Data Items** pane.
14. Pause the pointer over the page explorer button and click a report page.
15. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the report type (p. 79) you want to create to the work area.

    For example, drag a list, crosstab, chart, or repeater.
16. Click the data container.
17. In the **Properties** pane, click the select ancestor button and then click the container you just created.

    For example, if you created a list, click **List**.
18. Click the **Query** property and click the query you want to use for the report type.
19. In the **Insertable Objects** pane, on the **Data Items** tab, drag the items you want from the query you chose in the previous step to the data container.

Choose the Data Source

If you converted a query to SQL or MDX or created a query by adding your own SQL or MDX, you must choose the data source against which you want to execute the query.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. Click the **Data Source** object of the query for which you want to choose the data source.
3. In the **Properties** pane, double-click the **Data Source** property.
4. Click the data source you want and click **OK**.

First-Rows Optimization

The SQL produced by Report Studio depends on the report format you choose. For example, if you specify HTML format, first-rows optimization is requested. All-rows is requested if you specify PDF.

It is important for database administrators and programmers to remember that Report Studio does not always use first-rows optimization. If you assume first-rows optimization is always requested, this can cause the RDBMS optimizer to process the query differently than you expect.

Supported MDX Syntax

If you are working with MDX, you must be aware of the syntax that Report Studio supports. Report Studio supports the MDX grammar as specified in Microsoft's Data Access SDK version 2. 8, with the following exceptions.

Report Studio does not support the following MDX functions. This is a partial list and applies only when you are working with the following data sources:

- cubes
- DB2 OLAP
- Cognos Consolidation
- Cognos Contributor

For these data sources, Cognos 8 uses a variation of the MS Analysis Services syntax.

AllMembers	Ignore	SetToArray
Ascendants	Is	SetToStr
CalculationCurrentPass	IsAncestor	StddevP
CalculationPassValue	IsGeneration	StdevP
Call	IsLeaf	StripCalculatedMembers
Correlation	IsSibling	StrToMember
Covariance	LinkMember	StrToSet
CovarianceN	LinRegIntercept	StrToTuple
Current	LinRegPoint	StrToValue
DataMember	LinRegR2	ToggleDrillState
DrillDownLevel	LinRegSlope	TupleToStr
DrillDownLevelBottom	LinRegVariance	UniqueName
DrillDownLevelTop	LookupCube	UserName
DrillDownMember	MemberToStr	ValidMeasure
DrillDownMemberBottom	Name	VarianceP
DrillDownMemberTop	NameToSet	VarP

DrillupLevel	NonEmptyCrossjoin	VisualTotals
DrillupMember	Predict	

Add a Query Macro

Add a query macro to allow run time changes to be made to SQL queries. A macro is a fragment of code that you can insert in the Select statement of a query or in an expression. For example, add a macro to insert a new data item containing the user's name.

You can also add query macros to the Framework Manager model. For more information, including examples, see the Framework Manager *User Guide*.

Support for query macros in Report Studio includes the same capabilities as macros used in Framework Manager, including support for parameter maps defined within the Framework Manager model. However, Report Studio query macros do not extend to the layout. Therefore, when making changes to the query using macros, you must bear in mind the side-effects on the layout. For example, if a macro removes a column from the query that the layout refers to, a run-time error will occur.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. If the query to which you want to add a macro was built by using your own SQL, do the following:
   - Click the **SQL** object that is linked to the query.
   - In the **Properties** pane, double-click the **SQL** property.
   - In the **SQL** dialog box, type the macro.
   - Click **OK**.
3. If the query to which you want to add a macro was not built by using your own SQL, do the following:
   - Click the query.
   - In the **Properties** pane, double-click the **Generated SQL/MDX** property.
   - In the **Generated SQL/MDX** dialog box, click **Convert**.
   - In the **SQL** dialog box, type the macro.
   - Click **OK**.

Tip: For a list of supported macro functions, see "Using the Expression Editor" (p. 357).

Creating Prompts Using Query Macros

You can create mandatory and optional prompts in reports using query macros. Use the prompt macro functions `prompt` and `promptmany` to create single-value and multiple-value prompts. You

can use prompt macro functions when working with a relational data source or a dimensionally modeled relational (DMR) data source.

The `prompt` and `promptmany` functions have the following mandatory and optional parameters. All argument values must be specified as strings.

Parameter	Description
Name (mandatory)	Name of the prompt. Can also refer to the name of a parameter on a user-created prompt page, in which case the user-created prompt page appears when the report is run instead of the default prompt page that the macro would generate.
Datatype (optional)	Prompt value data type. The default value is string. Prompt values are validated. In the case of strings, the provided value is enclosed in single quotation marks and embedded single quotation marks are doubled.

Parameter	Description
	Values include: • boolean • date • datetime • decimal • double • float • int • integer • interval • long • memberuniquename -- Memberuniquename is not an actual data type. Use it in cases when the data type of the parameter is a member unique name (MUN). For example, use the syntax #prompt('Parameter', 'memberuniquename')#. • numeric • real • short • string • time • timeinterval • timestamp • token -- Token is not an actual data type. It is a way to pass SQL. A token does not pass values. • xsddate • xsddatetime • xsddecimal • xsddouble • xsdduration • xsdfloat • xsdint • xsdlong • xsdshort • xsdstring • xsdtime
DefaultText (optional)	Text to be used by default. If a value is specified, the prompt is optional. If you use a space and no values are provided in the **Prompt Value** dialog box, a `Where` clause is usually not generated. If you use text and no values are provided in the **Prompt Value** dialog box, a `Where` clause is usually generated using the default value. Ensure that the text you provide results in a valid SQL statement.

Parameter	Description
Text (optional)	Text that precedes any user-provided values, such as, 'and column1 = '.
QueryItem (optional)	The prompt engine can take advantage of the **Prompt Info** properties of the query item. Descriptive information can be shown although the prompt value is a code.
TextFollowing (optional)	The closing parenthesis which is used most often for the `promptmany` function. It is also useful when the prompt is optional and is followed by hardcoded filters in the SQL statement.

Example - Select a Country Prompt

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a report that will prompt users to choose the country for which they want to see data.

The following code shows how you can use macros to create a prompt:

```
select
 COUNTRY_MULTILINGUAL.COUNTRY_CODE
as COUNTRY_CODE,
 COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
 COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
 COUNTRY_MULTILINGUAL.CURRENCY_NAME
as CURRENCY_NAME
from
 gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#
```

Notes

- Because the `<Datatype>` argument is not specified, valid prompt values must be strings, which is correct in this case.
- Because the `<DefaultText>` argument is not specified, the `CountryName` prompt is a required prompt.

Chapter 6: Working with Existing Reports

After you have created a report, you can make changes or enhancements, such as setting up the report for bursting.

Before you modify an existing report, ensure that you have the proper security permissions. For more information, see the *Administration and Security Guide*.

You can

- open and save reports locally
- open a report from the clipboard
- open files from other Studios
- manage changes in the package
- add multiple items to a single column
- burst a report
- customize reports using conditions
- set up drill-through access
- drill-up/drill-down in the same report
- create a master detail relationship

Open and Save a Report Locally

By default, the reports you create are stored on the Cognos 8 server. You can, however, open and save reports on your computer. This is useful if you want to send a report to a report author that is working in a different environment, or you want to save the report to a source code controlled directory on a local network or drive.

Steps

1. Obtain the LFA.dll from your Cognos 8 administrator.

   The DLL is located in the bin directory where Cognos 8 is installed.

2. Open a command prompt window.

3. Register the LFA.dll file by typing

   regsvr32 LFA.dll

4. In Internet Explorer, set your computer and the Cognos 8 server as trusted sites:

- From the **Tools** menu, click **Internet Options**.
- On the **Security** tab, click **Trusted sites**.
- Click the **Sites** button.
- In the **Add this Web site to the zone** box, type **http://localhost** and click **Add**.

  Tip: If the Cognos 8 server is not on the same computer as the browser, type **http://<computer name>** instead.
- Repeat the above step for **http://<Cognos 8 servername>**.
- Clear the **Require server verification (https:) for all sites in this zone** check box.
- Click **OK** twice.

5. In Report Studio, from the **Tools** menu, click **Options**.
6. Select the **Allow local file access** check box and click **OK**.

   The menu items (**Local**) **Open** and (**Local**) **Save As** appear in the **File** menu.
7. Close and restart Report Studio.

You can now open and save reports on your computer. The first time you try to open or save a report locally, Internet Explorer asks you whether you want to allow an ActiveX control on the page to interact with other parts of the page. Click **Yes** to continue.

Open a Report from the Clipboard

You can open a report specification (p. 78) that was previously copied to the clipboard (**Tools** menu, **Copy To Clipboard**). This is useful for importing an XML report specification from outside the Cognos 8 environment.

Although Report Studio attempts to validate the report specification, it is your responsibility to ensure that it is correct. For more information, see the Cognos 8 SDK *Developer Guide*.

Step

- From the **Tools** menu, click **Open Report From Clipboard**.

Open Files from Other Studios

You can open reports, templates, or analyses that were created in Query Studio or Analysis Studio in Report Studio. All the capabilities of Report Studio are available to you, to change the formatting, layout, calculations, and queries.

You can also open a report, template, or analysis in Report Studio from Cognos Connection with the **Open with Report Studio** action.

Steps

1. From the **File** menu, click **Open**.
2. Click the report, template, or analysis that you want.
3. Click **Open**.

Note: If you make and save changes to a Query Studio report or an Analysis Studio analysis in Report Studio, the report or analysis can no longer be opened in Query Studio and Analysis Studio.

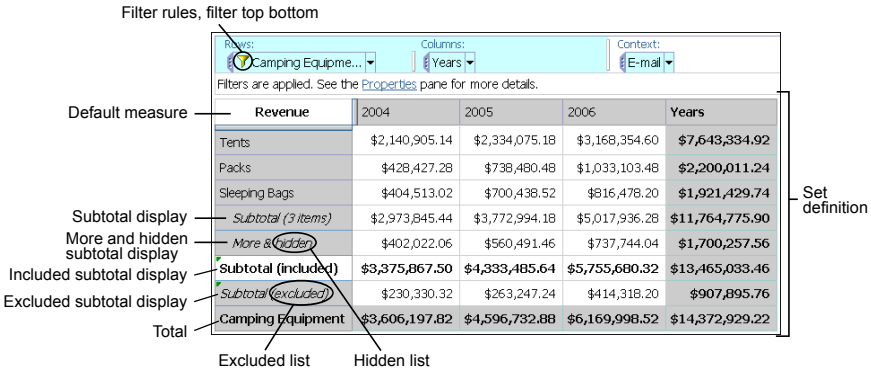
An Analysis Studio Query Specification

Analysis Studio defines each group of rows and columns as a set. When an analysis is imported into Report Studio, the report will have one query which processes all the sets found on the crosstab. Each set is defined by 18 data items which segment and summarize the base set definition. To maintain reports converted from Analysis Studio, you must understand what each of these items represent and how they relate to each other.

The data items for a set specify the following:

- set definition
- set segments, including which members were excluded and hidden individually
- filter rules for defining which members are to be retrieved
- calculations for Subtotal (N items), More & Hidden, Subtotal (included), Subtotal (excluded) , and the total

  For more information, see the Analysis Studio *User Guide*.
- which subtotals should appear
- any user-defined calculations



The default measure identifies which measure is used as the default for the crosstab or chart. If no default measure is specified, it may be empty.

Set Definitions

Data item	Definition	Dependencies
<*set name*> (base)	Defines the set of members to be used for sorting, filtering and summary operations. This data item provides a generic reference for all other data items and may reference one of the other set definitions.	None
<*set name*> (level)	Identifies the level for a level-based set.	None
<*set name*> (list)	Defines the list of members in a selection-based set.	None
<*set name*> (depth N)	Defines the set of members at N, number of levels down.	None
<*set name*> (named set)	References a predefined set.	None

Set Segments Definitions

Data item	Definition	Dependencies
<*set definition*> (hidden list)	Lists the members manually hidden using the **Hide** command in Analysis Studio. This set appears when the user tries to unhide a member.	<*set definition*>
<*set definition*> (included set)	Lists the set of members after filters are applied and hidden items are excluded, but before the **More** limit is applied. Sort or order operations, if any, are defined in this data item.	<*set definition*> (hidden list)

Data item	Definition	Dependencies
<*set definition*> (visible items set)	Limits the set to show the number of members according to the **More** limit, with a small tolerance. The tolerance allows for showing the last two members if that is all that remained in the **More** portion of the set. The choice of visible members can be based on a sort order.	<*set definition*> (included set)
<*set definition*> (excluded list)	Identifies those members that were manually excluded from the analysis. If no members were manually excluded, the expression defines an emptySet(). This set appears when the user tries to remove a member from the **Excluded items** list in the Properties pane.	<*set definition*>

Filters

Data item	Definition	Dependencies
<*set definition*> (filter rules)	Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than, based on measures, calculations or attributes.	<*set definition*>
<*set definition*> (excluded filters)	Removes those members that were manually excluded from the results after applying the user-defined rules.	<*set definition*> (excluded list), <*set definition*> (filter rules)

Data item	Definition	Dependencies
<*set definition*> (filter top bottom)	Focuses on the members based on top / bottom / first *n*, where *n* can be a count or a percentile. For more information about Top/ Bottom filters, see the Analysis Studio *User Guide*.	<*set definition*> (excluded filters) and totals for sets on the opposite axis
<*set definition*> (filter rules)	Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than, based on measures, calculations or attributes.	<*set definition*>

Subtotals and Related Conditions

Data item	Definition	Dependencies
<*set definition*> (subtotal)	Aggregates the visible items shown for the set.	<*set definition*> (visible items set)
<*set definition*> (subtotal display)	Shows the subtotal if the number of included items is greater than the number of visible items.	<*set definition*> (visible items set), <*set definition*> (included set), <*set definition*> (subtotal)
<*set definition*> (more and hidden subtotal)	Calculates the **More & hidden** subtotal by subtracting the aggregation of the included members from the aggregation of those members manually hidden or hidden by exceeding the maximum display limit. Not available for selection-based sets.	<*set definition*> (included set) , <*set definition*> (visible items set), <*set definition*>
<*set definition*> (more and hidden subtotal as set)	Converts the **More & hidden** subtotal member to a set for use in set operations.	<*set definition*> (more and hidden subtotal)

Data item	Definition	Dependencies
<*set definition*> (more and hidden subtotal as member)	Generically references **More & hidden** subtotal.	<*set definition*> (more and hidden subtotal)
<*set definition*> (more and hidden subtotal display)	Shows the **More & hidden** subtotal if the number of items hidden or clipped is greater than zero.	<*set definition*> (more and hidden subtotal)
<*set definition*> (included subtotal)	Calculates the value for **Subtotal (included)**. For selection-based sets, this data item references the summary data item for the set.	<*set definition*> (included set)
<*set definition*> (included subtotal display)	Shows **Subtotal (included)** if any members passed the filter criteria.	<*set definition*> (included set)
<*set definition*> (excluded subtotal)	Calculates the value for **Subtotal (excluded)**. This data item is not available for selection-based sets.	<*set definition*> (total), <*set definition*> (included subtotal), <*set definition*>
<*set definition*> (excluded subtotal display)	Shows **Subtotal (excluded)** if the number of members in the filtered set is less than the base set. The filter rule is filtering out something.	<*set definition*> (excluded subtotal), <*set definition*>

Data item	Definition	Dependencies
<*set definition*> (total)	Identifies the total for the set and is calculated directly from the data store. The expression used to calculate the summary depends on the set type. Details-based set: The selected member. Level-based and depth-based sets: The root member for the hierarchy. Selection-based set: The aggregate, typically the sum, of the selected members.	<*set definition*>

Managing Changes in the Package

If changes were made to the package used to create a report, the report must be updated. When you open a report, Report Studio automatically checks to see if the package has changed. If it has, a message appears indicating that the report will be updated to the latest version of the package. However, you may need to make additional changes to the report if

- the namespace name or query subject and data item names in the package changed
- the name of the package changed

Update Name References

If the namespace name or query subject and data item names in the package have changed, you must update reports created with the package to reflect the change.

The names of data items in a report are a concatenation of the namespace name, query subject name, and the data item name. For example, if you add Order number from the GO Data Warehouse (query) sample package to a filter expression, you will see [Sales (query)].[Sales order].[Order number] in the expression. Similarly, package filter names are a concatenation of the namespace name and the filter name.

Steps

1. To update a data item name reference:
   - Click the data item.
   - In the **Properties** pane, double-click the **Expression** property.

- In the **Expression Definition** box, update the data item name reference, and click **OK**.

2. To update a package filter name reference:
   - From the **Data** menu, click **Filters** .
   - Double-click the package filter.
   - In the **Expression Definition** box, update the namespace name, and click **OK**.

Change the Package

If the name of the package used to create a report has changed, change the package connection to update the report.

At the same time, you can also change the authoring language for the report.

Steps

1. From the **File** menu, click **Report Package**.
2. Click the **Another package** radio button and choose the package that you want.
3. To change the authoring language, click the ellipsis (...) button beside the **Language** box, choose a different report language, and click **OK**.

   You may need to update the report to reflect the change. For example, any filter expressions in the report must be modified to reflect the syntax rules of the new language.
4. Click **OK**.

   The new package is loaded and the report is validated. If errors are found, the **Validation Errors** dialog box appears, showing what elements must be updated to reflect the package change.
5. Click **Close**.
6. Make any required changes in the report to support the new package.

   For example, you may need to link the data items in the report to the new package.

   Tip: You can use Query Explorer (p. 213) to make all the changes.

Add Multiple Items to a Single Column

You can add multiple items to a single column to condense a report. For example, you have a list report that contains many columns. You can reduce the number of columns in the list by putting related information in a single column.

Steps

1. From the **Structure** menu, click **Lock Page Objects** to unlock the report.
2. In the **Insertable Objects** pane, drag the items that you want to the column.

For example, you can add data items from the **Source** tab, or text items from the **Toolbox** tab.

Example - Create a Report with Multiple Items in One Column

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a list report showing the name, position, email address, phone number, extension, and fax number for each sales representative in each city, organized by country. To reduce the number of columns in the list, you show email addresses, phone numbers, extensions, and fax numbers in a single column.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab, expand **HR (query)**, **Employee summary (query)**, and **Employee by region**. Add the following data items to the list by double-clicking them:
   - **Country**
   - **City**
   - **Employee name**
   - **Position name**
   - **Email**
   - **Work phone**
   - **Extension**
   - **Fax**
6. Click the **Country** column, and from the **Structure** menu, click **Section**.
7. From the **Structure** menu, click **Lock Page Objects**.

   The report is unlocked.
8. Click **Country**, and in the **Properties** pane, double-click the **Font** property.
9. Change the font to **Arial Black**, **11 pt**, and **Bold**, and click **OK**.
10. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Table** to the right of the **Work phone** text item in the **Work phone** column (not the column title), and create a table that has one column and three rows.

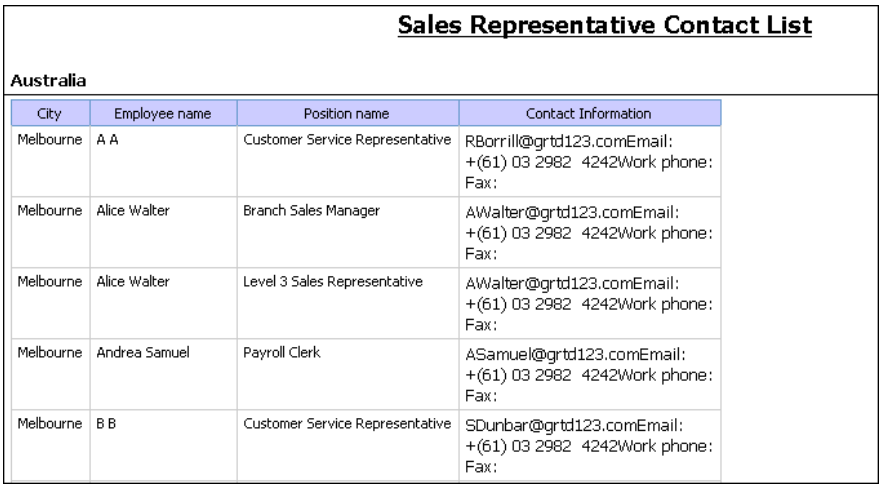
11. Drag the following data items to the table:
    - **Email** to the first row in the table
    - **Work phone** to the second row in the table
    - **Fax** to the third row in the table
12. Drag **Extension** to the right of **Work phone**.
13. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Text Item** to the left of each item in the table, and type the following for each item, putting a blank space before and after the text:
    - **Email:**
    - **Work phone:**
    - **ext:**
    - **Fax:**
14. Press Ctrl+click to select the **Extension**, **Email**, and **Fax** column titles, and click the delete button .
15. Click the **Work phone** column title.
16. In the **Properties** pane, click the **Source Type** property and click **Text**.
17. Double-click the **Text** property.
18. Type the following and click **OK**.

    Contact Information
19. Double-click the text item in the page header, type the following, and click **OK**.

    Sales Representatives Contact List
20. From the **Structure** menu, click **Lock Page Objects**.

    The report is locked.
21. Save the report.

When you run the report, contact information for each sales representative appears in a single column.



Sales Representative Contact List

Australia

City	Employee name	Position name	Contact Information
Melbourne	A A	Customer Service Representative	RBorrill@grtd123.comEmail: +(61) 03 2982 4242Work phone: Fax:
Melbourne	Alice Walter	Branch Sales Manager	AWalter@grtd123.comEmail: +(61) 03 2982 4242Work phone: Fax:
Melbourne	Alice Walter	Level 3 Sales Representative	AWalter@grtd123.comEmail: +(61) 03 2982 4242Work phone: Fax:
Melbourne	Andrea Samuel	Payroll Clerk	ASamuel@grtd123.comEmail: +(61) 03 2982 4242Work phone: Fax:
Melbourne	B B	Customer Service Representative	SDunbar@grtd123.comEmail: +(61) 03 2982 4242Work phone: Fax:

Bursting Reports

Burst a report to distribute its contents to various recipients. Bursting is the process of running a report once and then dividing the results for distribution to recipients who each view only a subset of the data. For example, salespeople in different regions each need a report showing the sales target for each country. You use burst reports to send each salesperson the information they need. Burst reports can be distributed by email or saved to a directory for viewing in Cognos Connection.

To burst a report against a dimensional data source, see "Creating Burst Reports Using a Dimensional Data Source" (p. 251).

You cannot burst crosstab reports.

To create a burst report, you

- define the burst recipients
- specify burst groups
- set burst options
- enable bursting

Defining Burst Recipients

Define the recipients that will receive data when the report is run. You can distribute burst reports to individual users, groups, roles, distribution lists, and contacts. To define the recipients, you can

- create a calculated field in the report
- create a burst table in the data source

Create a Calculated Field

You can use a calculated field to dynamically create the burst recipients.

Steps

1. Pause the pointer over the query explorer button and click the query that will produce the data you want to distribute.
2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Items** pane.
3. In the **Expression Definition** box, type the expression that will generate the list of recipients and click **OK**.

   For example, the following expression builds the list of the employees of The Great Outdoors Company. The expression concatenates the first letter of each employee's first name with their last name.

   ```
   lower(substring([Employee summary (query)].[Employee by organization].[First name],1,1) + [Employee summary (query)].[Employee by organization].[Last name])
   ```

4. To give the data item a more meaningful name, in the **Properties** pane, click the **Name** property, type a different name, and press the Enter key.

Creating a Burst Table

You can create a burst table to specify the list of recipients.

You can also use an existing table as the burst table.

Creating a burst table involves

- creating the table in the source database
- importing the table into a package

Create the Burst Table in the Source Database

Create a table in the source database for the list of recipients. The steps you must follow depend on the database system you are using. The burst table must contain the following columns:

- A unique identifier.

  Tip: Some database systems do not require a unique identifier for each table.

- A recipient column.
- The data item that you want to burst on.

You can also include other columns that provide additional information. For example, if you plan to distribute reports by email, you can add a column that will contain the email address of each recipient.

After you create the table, add the recipients you want to receive the report. You can create a mixed recipients list that includes individual users, groups, roles, contacts, distribution lists, or email addresses. For example, a burst table may contain the following recipients.

Recipient example	Recipient type
CAMID(":Canada")	Group
CAMID(":")/contact[@name='Silvano Allessori']	Contact
CAMID(":")/distributionList[@name='European Partners']	Distribution list
CAMID("LDAP_Local_ID:u:uid=gbelding,ou=people")	Authentication provider user or group, where LDAP_Local_ID is the name of an LDAP namespace ID, and people is the name of an organizational unit
c8@cognos99.com	Email address

CAMID stands for Cognos Access Manager ID, and it represents an internal search path to the recipients. Specify search paths when you want to save burst reports in a directory. You can obtain the path in Cognos Connection. Open the **Set properties** page for each recipient, and click the **View the search path** link. Ensure that you use the proper syntax when adding recipients to the burst table.

In the case of NTLM namespaces, user IDs in the search path use alphanumeric or numeric characters that make them difficult to read. You can use the following alternate search path syntax:

```
directory/namespace[@name="Local NT"]//account[@userName="gbelding"]
```

where Local NT is the name of a namespace and gbelding is the name of a user. The double slash before the account element indicates that you are searching all accounts under the specified namespace.

Note: If you have a mixed recipients list, do not mix email address recipients and alternate path recipients. Because the alternate path syntax contains the @ symbol, it will be mistaken for an email address.

For more information about users, groups, roles, contacts, and distribution lists, see the *Administration and Security Guide*.

Import the Table

After you create the burst table in the source database, you must add it to the package that you will use to create the report.

In Framework Manager, do the following:

- Open the package you want.
- Import the table.

- Define the relationship between the burst table and the table containing the data item that you want to burst on.

  For example, you are bursting on country code. You define a relationship between country code in the burst table and country code in the Country table.

- Save and publish the package.

For more information about importing tables and creating relationships, see the Framework Manager *User Guide*.

Specify Burst Groups

Specify burst groups to set how the report is distributed. Burst groups are defined by a data item that you create in the report or that you add from the burst table.

Steps

1. Pause the pointer over the query explorer button and click the query that will produce the data you want to distribute.
2. If you are creating a data item, do the following:
   - In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Data Item** to the **Data Items** pane.
   - In the **Expression Definition** box, type the expression that defines the burst key and click **OK**.

     For example, the following expression builds an email address for each sales representative in The Great Outdoors Company. The expression incorporates the calculated field previously created, named `userID` below, with `cognos99.com` as the domain name.

     ```
     [userID]+'@cognos99.com'
     ```

     Tip: To give the data item a more meaningful name, in the **Properties** pane, click the **Name** property, type a different name, and press Enter.
3. To specify a burst table column as the data item, do the following:
   - In the **Insertable Objects** pane, on the **Source** tab, expand the burst table.
   - Drag the data item you want to use to the **Data Items** pane.

     For example, if you are bursting reports by email, drag the data item containing email addresses.

Set Burst Options

Set burst options for the report to indicate what data item to burst on and who the recipients are.

Before you set burst options, ensure that the data item you intend to burst on is in the report and grouped. The grouped column will create the appropriate subsets of data. In addition, you must associate the burst key with this level of grouping.

Steps

1. From the **File** menu, click **Burst Options**.
2. Select the **Make report available for bursting** check box.
3. Under **Burst Groups**, in the **Query** box, click the query that contains the data item you want to burst on.

   Tip: You can choose a query that does not appear in the layout. This is useful when you want to distribute the same report to all burst recipients.
4. In the **Label** box, click the data item to be used to label each burst report.
5. Click the edit button .
6. In the **Data Items** box, drag the data item that you want to burst on to the **Groups** folder and click **OK**.

   Tip: You can specify the sort order of data within each group by dragging data items to the **Sort List** folder and then clicking the sort order button .
7. Under **Burst Recipient**, in the **Query** box, click the query that contains the data item to be used as the distribution list.
8. In the **Data Item** box, click the data item that contains the recipients.
9. In the **Type** box, choose the method to use to burst the report.
   - Click **Email addresses** to distribute reports by email.
   - Click **Directory entries** to distribute reports to a directory that recipients can access in Cognos Connection.

     When a recipient logs into Cognos 8, he will see only the report that is specific to him.
   - Click **Automatic** to let Cognos 8 determine from the data item whether to email reports or send them to a directory.
10. If the report contains two nested data containers, such as a list and a chart, click the ellipsis (...) button beside **Master Detail Relationships**, and define the relationship between the containers.

    For information about master detail relationships, see "Create a Master Detail Relationship" (p. 282).
11. Click **OK**.

Enable Bursting

When the report is ready to be distributed, enable bursting for the report in Cognos Connection.

Steps

1. Locate the report in Cognos Connection.
2. Under **Actions**, click **Run with options**.
3. Click the **Advanced options** link.
4. Under **Time and mode**, select **Run in the background**.
5. Select the **Burst the report** check box.
6. If you are distributing reports by email, select the **Send the report by email** check box.

   Tip: If you are bursting the report to a directory, you can also send the report by email if the check box is selected. Reports will be emailed if the recipient's email address is stored in the authentication source you are using, or if it is entered in the recipient's personal information in Cognos Connection.
7. If the burst report contains a drill-through link (p. 273) to another report and you are distributing the report by email, do the following:
   - Click the **Edit the email options** link.
   - Select the **Include a link to the report** check box and click **OK**.

   If you do not select the check box, the drill-through links in the burst report will not work.
8. Click **Run** to run the report.

Allow a few moments for the report to run. If you are an administrator, you can view all outputs for the report. Under **Actions**, click **View the output versions for this report**. When burst recipients log into Cognos Connection or access their email accounts, they will see only the data that is meant for them.

Creating Burst Reports Using a Dimensional Data Source

You can burst a report using a dimensional data source by using burst information stored in the data source. We assume that you do not want to append bursting information to existing dimensional data sources. The solution is to create a relational data source that contains the burst information.

Report bursting is limited when the underlying data source is a cube (MOLAP data source such as Cognos PowerCube, Microsoft Analysis Services, Hyperion Essbase, or IBM DB2/OLAP). The burst report must be a grouped report and the burst is restricted to the outermost grouping in the report. For example, if you have a report grouped on Country and State, then you can burst the report only on Country.

To burst a report using a dimensional data source, do the following:

- In Framework Manager, include both the dimensional data source that is the basis for reporting and the relational burst table in the model.

  For more information about models, see the Framework Manager *User Guide*.

- In Report Studio, create a master detail report in which the master query drives the report of interest and the detail query contains the burst information.

  The master query must be grouped by a data item that you are bursting on and that has a corresponding data item in the relational burst table.

  Author the detail query against the relational burst table. The burst table must contain two columns, the data item corresponding to the one used in the master report for bursting, and the data item that contains the recipient information. The recipient can be an email address or an expression that results in a search path to an object in Content Manager, such as an account, group, role, contact, or distribution list.

  For more information about master detail queries, see "Create a Master Detail Relationship" (p. 282).

- Ensure that the detail query, which must be evaluated by Cognos 8 when the report is executed, is not visible:

  - Place a list that is based on the detail query in a conditional block for which the box type is set to **None**.
  - Link the master and detail queries using the following expression:

    ```
    [Master Burst Key] = [Detail Burst Key]
    ```

When you set the burst options for the report, the master query provides the data items for the burst key, and the detail report provides the data items for the burst recipients.

Example - Burst a Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales for each sales representative. The report is to be emailed to each sales representative, but they do not need to see data for everyone, only the data that pertains to them. You create a list report that you burst to each sales representative.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Data Warehouse** (**query**) link, click the **Launch** link in the upper right corner of the page, and then click **Report Studio**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and marketing** (**query**) and **Sales** (**query**). Add the following data items by double-clicking them:

- **Employee name**
- **Product line**
- **Product type**
- **Product name**
- **Quantity**
- **Revenue**

Tip: You can find these data items in the **Employee by organization**, **Product** and **Sales fact** folders.

6. Group the **Employee name**, **Product line**, and **Product type** columns.
7. Click the **Employee name** column, and then click the create header button.

   Employee name appears as a header in the list. You no longer need to keep the data item as a list column.
8. In the list, click **Employee name** and click the delete button .
9. Click **Revenue**, click the aggregate button, and click **Total**.
10. Pause the pointer over the query explorer button and click **Query1**.
11. In the **Insertable Objects** pane, on the **Source** tab, drag **Email** from the **Employee by organization** folder to the **Data Items** pane.
12. From the **File** menu, click **Burst Options**.
13. Select the **Make report available for bursting** check box.
14. Under **Burst Groups**, in the **Query** box, click **Query1**.
15. In the **Label** box, click **Employee name**.
16. Click the edit button .
17. In the **Data Items** box, drag **Employee name** to the **Groups** folder and click **OK**.
18. Under **Burst Recipient**, in the **Query** box, click **Query1**.
19. In the **Data Item** box, click **Email**.
20. In the **Type** box, click **Email addresses**.
21. Click **OK**.
22. Save the report.
23. Locate the report in Cognos Connection.
24. Under **Actions**, click **Run with options** .

25. Click the **Advanced options** link.
26. Select the **Burst the report** check box.
27. Select the **Send the report by email** check box.
28. Click **Run** and then click **OK**.

When sales representatives access their email accounts, they will see a report with only the data that is meant for them.

Customizing Reports Using Conditions

You can define conditions to control what users see when they run a report. Conditions can apply to specific items in a report. For example, you can define a conditional style to highlight exceptional data, such as product revenue that exceeds your target.

Conditions can also apply at the report layout level. Conditional layouts are useful for delivering reports to a multilingual audience. For example, you can have text items, such as titles and cover pages, appear in the same language as the data in the report.

You can use conditions to

- highlight data using conditional styles
- highlight data using style variables
- specify which objects are rendered when a report is run

Highlight Data Using Conditional Styles

Add conditional styles to your report to better identify exceptional or unexpected results. A conditional style is a format, including such things as cell shading or font color, that is applied to objects if a specified condition is true.

For example, you want to automatically highlight in green departments in your organization that meet their budget quotas and highlight in red the departments that go over budget. Creating conditional styles quickly and easily color-codes information in your reports so that you can find areas that need attention.

You can apply multiple conditional styles to objects. For example, you can apply one style for data formats in specific cells and another style for the overall report. If multiple styles set the same property, such as font color, the last style in the list is applied.

You can also create a conditional style once and reuse it on multiple objects in your report. And you can specify the order in which styles are applied. In addition, you can use existing local classes as your conditional styles.

You can create the following types of conditional styles.

Type	Description
Numeric Range	Highlights straight numerical data, such as revenues and losses.
Date/Time Range	Highlights data from specific dates and times.
Date Range	Highlights data from specific dates. For example, you can highlight sales data for specific periods of a fiscal year.
Time Range	Highlights data from specific times.
Interval	Highlights data falling between set intervals.
String	Highlights specific alphanumeric items in a report. For example, you may want to highlight all instances of a specific word or phrase, such as Equipment. String criteria are case-sensitive. If multiple string conditions are met, only the style of the first condition is applied.
Advanced	Creates conditional styles that use calculations or expressions. If multiple advanced conditions are met, only the style of the first condition is applied. For more information about creating conditional styles using variables, see "Highlight Data Using Style Variables" (p. 257).

You can perform a search to find objects in your report that use conditional styles (p. 57). You can also view all the conditional styles used in your report so that you can delete or modify them (p. 257).

You can also use variables to highlight data (p. 257). If a report contains both conditional styles and style variables, the style variables are applied first and then the conditional styles are applied.

Steps to Create a New Conditional Style

1. Click the object for which you want to define a conditional style, and then click the conditional styles button .

   Tip: You can also right-click the object and click **Style**, **Conditional Styles**, or in the **Properties** pane, double-click the **Conditional Styles** property.

2. Click the add button and click **New Conditional Style**.

3. Select the data item that you want to use to determine the condition.

   Depending on the type of data item you select, Report Studio selects the type of conditional style that you can use.

4. Type a name for the conditional style.

5. If you want to define a value, date/time, date, time, or interval condition, do the following:
   - Click the new button and type or select a value to define a threshold.

     The value appears under the **Range** column, and two ranges are created.
   - For each range, under **Style**, click one of the predefined styles that you want to apply to the range, or click the edit style button and create a new style.

     Tip: You can also define a style for the cells in your report that have missing values.
   - Repeat to add other values.

     Tip: Under **Style**, pause the pointer over each range to see the condition produced for each range.
   - If you want to move a value above or below a threshold, click the arrow button next to the value.

     For example, you insert a threshold value of five million. By default, the ranges are less than or equal to five million and greater than file million. Moving the five million value above the threshold changes the ranges to less than five million and greater than or equal to five million.
6. If you want to define a string condition, do the following:
   - Click the new button and select how to define the condition.
   - If you want to select one or more individual values, click **Select multiple values** and click the values you want.
   - If you want to type specific values, click **Enter values** and type the values you want.
   - If you want to specify your own criteria, such as begins with the letter A, click **Enter string criteria** and specify your condition.
   - For each condition, under **Style**, click one of the predefined styles that you want to apply, or click the edit style button and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside **Remaining values (including future values)**.
   - Specify the order in which to evaluate the conditions.

     Conditions are evaluated from top to bottom and the first condition that is met is applied.

Steps to Reuse an Existing Conditional Style

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button.

   Tip: You can also right-click the item and click **Style**, **Conditional Styles**, or in the **Properties** pane, double-click the **Conditional Styles** property.

2. Click the add button, click **Use Existing Conditional Style**, and select the style you want to use.

Steps to Create an Advanced Conditional Style

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button.

   Tip: You can also right-click the item and click **Style**, **Conditional Styles**, or in the **Properties** pane, double-click the **Conditional Styles** property.

2. Click the add button and click **Advanced Conditional Style**.
3. Type a name for the conditional style.
4. Click the new button and specify the expression that defines the condition.
5. For each condition, under **Style**, click one of the predefined styles that you want to apply, or click the edit style button and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside **Remaining values (including future values)**.
6. Specify the order in which to evaluate the conditions.

   Conditions are evaluated from top to bottom and the first condition that is met is applied.

Manage Conditional Styles

If your report uses conditional styles, you can easily see all the ones that are applied to your report. You can then easily globally modify or delete them. You can also define a new conditional style that you can make available for objects in your report.

In the Report Studio options, you can specify whether to automatically delete conditional styles that are no longer used in a report (p. 57).

Step

- From the **Tools** menu, click **Manage Conditional Styles**.

Highlight Data Using Style Variables

Highlight data in your report to better identify exceptional results. For example, you want to identify sales representatives who have exceeded their quota. You create a condition that checks whether each representative's sales for the year is greater than their quota for the year.

Use style variables if you are working with reports created in a previous version of Cognos 8 or if you want to use language variables to specify conditional styles.

You can also use conditional styles to highlight data (p. 254). If a report contains both conditional styles and style variables, the style variables are applied before the conditional styles.

Steps

1. Create a variable and define the condition that determines if the data will be highlighted.
2. In the work area, click the column that you want to highlight based on the condition you created.
3. In the **Properties** pane, double-click the **Style Variable** property.
4. Click **Variable**, click the variable you want to assign to the object, and click **OK**.
5. If you assigned a string variable, in the **Values** box, select the values you want the condition to support.

   Tip: A default value exists for the variable, and it is always selected.
6. If you assigned a language variable, in the **Values** box, select the languages you want the condition to support.

   Tip: A default value exists for the variable, and it is always selected.
7. Click **OK**.
8. Pause the pointer over the condition explorer button , and click one of the possible values for the variable other than the default value.

   Tip: When you select a value in **Condition Explorer**, the Explorer bar becomes green to indicate that conditional formatting is turned on, and that any changes you make to the report applies only to the variable value.

   For example, if you created a boolean variable, click the **Yes** value.
9. In the **Properties** pane, specify the formatting that you want to highlight the column with when the condition is satisfied.

   For example, click the **Border** property to create a thicker border around the column.
10. Repeat steps 9 to 10 for other possible values defined for the variable.

    Tip: When pausing the pointer over the condition explorer button, click (**No variable**) to view how the report looks when no variable is applied. Or you can triple-click the Explorer bar.

When you run the report, the report objects to which you applied the variable are highlighted when the condition is satisfied. For example, if you created a boolean variable, the objects are highlighted when the condition is met. If the condition is not satisfied for any object, no conditional formatting is applied.

Adding Conditional Rendering to a Report

You can specify which objects are rendered when a report is run. You can:

- add a variable

- hide and show objects
- add conditional rendering
- add multiple layouts
- create multilingual reports

Add a Variable

Before you can add conditional formatting or conditional rendering to your report, you must add a variable in the condition explorer or in the **Properties** pane.

Steps in the Condition Explorer

1. Pause the pointer over the condition explorer button and click **Variables**.
2. In the **Insertable Objects** pane, drag one of the following variables to the **Variables** pane:
   - To create a variable that has only two possible values, **Yes** and **No**, drag **Boolean Variable**.
   - To create a variable whose values are string-based, drag **String Variable**.
   - To create a variable whose values are different languages, drag **Report Language Variable**.
3. If you created a boolean variable, in the **Expression Definition** box, define the condition and click **OK**.

   For example, the following expression returns the value Yes if revenue is less than one million dollars and the value No if revenue is greater than or equal to one million.

   [Revenue]<1000000

   For information about creating expressions, see "Using the Expression Editor" (p. 357).
4. If you created a string variable, do the following:
   - In the **Expression Definition** box, define the condition and click **OK**.

     For example, the following expression returns the value high if revenue is greater than one million dollars and the value low if revenue is less than or equal to one million dollars.

     `if` ([Revenue]>1000000) `then` ('high') `else` ('low')

     For information about creating expressions, see "Using the Expression Editor" (p. 357).
   - Click the add button under the **Values** pane.
   - For each value that the variable can assume, type the name of the value that corresponds with the possible outcomes defined in the expression.
   - Click **OK**.

     For example, in the previous expression, you must create two values for the variable, high and low.

Tip: You can create a group by clicking two or more values and then clicking the group values button.

5. If you created a language-specific variable, in the **Languages** dialog box, select the languages you want to support and click **OK**.

   Tip: You can create a group by clicking two or more values and then clicking the group values button. For example, you can create a group that includes all of the different French languages available.

Steps in the Properties Pane

1. Select the object that you want.
2. In the **Properties** pane, under **Conditional**, double-click the conditional property to which you want to assign the variable.

Goal	Conditional Property
Specify a variable based on which text can be conditionally shown. For example, you want different text to appear when a report is run in a different language.	**Text Source Variable**
Specify a variable based on which object can be conditionally rendered. For example, you want to make a revenue report smaller by not rendering rows that are below a threshold.	**Render Variable**
Specify a variable based on which object can be conditionally styled. For example, you want data that meets some criterion to appear in a different color.	**Style Variable**
Specify a variable based on which objects inserted into a block can be conditionally rendered. Applies only to conditional block objects that you insert in a report (p. 143).	**Block Variable**

3. Click **Variable** and click one of the following variable types:
   - **<New language variable>**
   - **<New string variable>**
   - **<New boolean variable>**

   Tip: To use an existing variable, you can select it here.

4. In the **New Variable** dialog box, type the name of the variable in the **Name** box.
5. If you created a string variable, click the add button and type the string values you want to define, and click **OK**.
6. If you created a language variable, select the languages you want to support, and click **OK**.
7. In the **Expression Definition** box, define the condition and click **OK**.

Hide and Show Objects

You can hide and show objects in a report based on a condition you define.

You can also specify that an object should not be rendered based on a condition (p. 261).

Steps

1. Create a variable and define the condition that determines if the object is to be shown or hidden.

   Tip: Create a boolean variable to show and hide objects, as this type of variable has only two possible values.

2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag the **Conditional Blocks** object to the work area.
3. Click the conditional block.
4. In the **Properties** pane, double-click the **Block Variable** property.
5. Click **Variable**, click the variable you created, and click **OK**.
6. Click the **Current Block** property and click **Yes**.
7. In the **Insertable Objects** pane, drag the object you want to show or hide in the conditional block.

   For example, drag a data item from the **Source** tab or from the **Data Items** tab.

   You may need to link the report page to a query (p. 151) before you can add a data item to the block.

When you run the report, the report objects to which you applied the variable are visible when the condition is satisfied and invisible when it is not.

Add Conditional Rendering

Add conditional rendering to specify which objects are rendered when a report is run. This is useful when your report contains sensitive data.

Conditional rendering is not the same as showing or hiding objects. When you hide an object, the object exists but is transparent. If an object is not rendered, it is not in the report.

For a list of objects that can be rendered conditionally, see the Render Variable property in "Report Studio Object and Property Reference" (p. 523).

Steps

1. Select the list column to be rendered conditionally.

   Tip: You must select the list column, not list column body or list column title. If the body or title is selected, as indicated in the **Properties** pane, use the select ancestor button to select the list column.

2. In the **Properties** pane, double-click the **Render Variable** property.

3. Click **Variable** and click the variable that will be used to determine if the column is to be rendered.

4. In the **Render for** box, select the values you want the condition to support and click **OK**.

   Tip: A default value exists for the variable, and it is always selected.

Example - Create a Conditional Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows orders after a date specified by the user. The report will prompt the user for a date, and will also ask whether the user wants to see a description for each order.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.

2. Click the **GO Data Warehouse (query)** link, click the **Launch** link in the upper-right corner of the page, and then click **Report Studio**.

3. In the **Welcome** dialog box, click **Create a new report or template**.

4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the **Source** tab, expand **Sales and Marketing (query)**, and **Sales (query)**. Add the following data items to the list:

   - **Date** (in **Time dimension**)
   - **Order number** (in **Sales order**)
   - **Product name** (in **Product**)
   - **Product description** (in **Product**)
   - **Quantity** (in **Sales fact**)
   - **Unit price** (in **Sales fact**)
   - **Revenue** (in **Sales fact**)

6. Click **Date** and click the section button .
7. Group the **Order Number** column by selecting the column and clicking the group button.
8. Click **Revenue**, click the aggregate button , and click **Total**.
9. Change the title of the report to **New Orders**.
10. Pause the pointer over the page explorer button and select **Prompt Pages**.
11. Create a new prompt page by double-clicking **Page** in the **Insertable Objects** pane.
12. Double-click the new prompt page.
13. In the **Insertable Objects** pane, on the **Toolbox** tab , double-click **Text Item** and type the following text:

    Enter the start date, and select if descriptions will be shown.
14. Insert a 2 by 2 table into the prompt page using the insert table button, and moving the pointer until four squares are highlighted in a 2 by 2 pattern.
15. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Text Item** into the upper-left cell and type the following text:

    Starting Date
16. In the **Insertable Objects** pane, drag a **Text Item** into the lower-left cell, and type the following text:

    Show Descriptions
17. In the **Insertable Objects** pane, drag a **Date Prompt** into the upper-right cell.
18. In the **Prompt Wizard** window, select **Create a new parameter** and type **p_Date** in the space provided, then click **Next**.
19. When prompted in the **Create Filter** window, select **Create a parameterized filter** with the following entries:
    - For **Package item**, click the ellipsis (...) button and click **[Sales (query)].[Time dimension].[Date]**.
    - For **Operator**, click **>**.
20. Click **Finish**.
21. In the **Insertable Objects** pane, drag a **Value Prompt** into the lower-right cell.
22. When prompted with the **Prompt Wizard**, **Choose Parameter** window, select **Create a new parameter** and type **p_ShowDesc** in the space provided, and then click **Finish**.
23. Select the **Value Prompt**, and in the **Properties** pane, double-click **Static Choices**.
24. Click the add button .

25. In the **Edit** dialog box, type **Yes** in both the **Use** and **Display** boxes.
26. Click the add button.
27. In the **Edit** dialog box, type **No** in both the **Use** and **Display** boxes.
28. Click **OK**.
29. Pause the pointer over the condition explorer button and click **Variables**.
30. Create a new boolean variable by double-clicking **Boolean Variable** in the **Insertable Objects** pane.
31. In the **Report Expression** dialog box, type the following in the **Expression Definition** window:

    ParamDisplayValue("p_ShowDesc") = 'Yes'
32. Click **OK**.
33. Click the new boolean variable that you created and in the **Properties** pane, for the **Name** property, type **showDesc**.
34. Pause the pointer over the page explorer button and click the report page.
35. Click the **Product descriptions** column.
36. In the **Properties** pane, select the list column by clicking the select ancestor button and selecting **List Column** from the context menu.
37. In the **Properties** pane, double-click the **Render Variable** parameter and select the **showDesc** boolean variable you created in steps 24 to 33.
38. Click **Run**.

The report will prompt you for a date, and will then provide orders that occur after the date you entered. The report will also ask whether to show the **Descriptions** column, and the column will be rendered only if you choose **Yes** to this selection.

Add Multiple Layouts

Add multiple layouts to show a report in different ways. For example, you can define a different layout for each language in a multilingual report. This allows you to create a single report that can be viewed by report consumers that use different regional settings.

Steps

1. Create a variable and define the condition that will be used for each layout.

   For example, create a report language variable that includes each language that requires a conditional layout.

   Note: Expressions used in a conditional layout cannot reference a query.
2. From the **File** menu, click **Conditional Layouts**.
3. Select a variable, and then select the values that require a separate layout.

4. Click **OK**.

A layout is created for each value you selected. Pause the pointer over the page explorer button to navigate the different layouts. For each layout, click the **Report Pages** link to create a report page, or click the **Prompt Pages** link to create a prompt page (p. 199), and add the objects you want.

Tip: You can create new variables from the **Conditional Layouts** dialog. The variables are added to the condition explorer. For more information see, "Add a Variable" (p. 259).

Creating Multilingual Reports

You can create reports that show data in more than one language and use different regional settings. This means that you can create a single report that can be used by report consumers anywhere in the world.

The samples databases provided with Cognos 8 store a selection of text fields, such as names and descriptions, in more than 25 languages to demonstrate a multilingual reporting environment. For information about how data is stored in the samples databases and how the samples databases are set up to use multilingual data, see the *Administration and Security Guide*.

Here is the process for creating a multilingual reporting environment:

- Use multilingual metadata.

  The data source administrator can store multilingual data in either individual tables, rows, or columns.

  For more information about configuring your database for multilingual reporting, see the *Administration and Security Guide*.

- Create a multilingual model.

  Modelers use Framework Manager to add multilingual metadata to the model from any data source type except OLAP. They add multilingual metadata by defining which languages the model supports, translating text strings in the model for things such as object names and descriptions, and defining which languages are exported in each package. If the data source contains multilingual data, modelers can define queries that retrieve data in the default language for the report user.

  For more information, see the Framework Manager *User Guide*.

- Create multilingual maps.

  Administrators and modelers use a Windows utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

  For more information, see the Map Manager *Installation and User Guide*.

- Create a multilingual report.

  The report author uses Report Studio to create a report that can be viewed in different languages. For example, you can specify that text, such as the title, appears in German when the report is

opened by a German user. You can also add translations for text objects, and create other language-dependent objects.

For more information, see the Report Studio Professional Authoring *User Guide*.

- Specify the language in which a report is viewed.

  You can use Cognos Connection to do the following:

  - Define multilingual properties, such as a name, screen tip, and description, for each entry in the portal.
  - Specify the default language to be used when a report is run.

    Tip: You can specify the default language on the run options page, in the report properties, or in your preferences.
  - Specify a language, other than the default, to be used when a report is run.

  For more information, see the Cognos Connection *User Guide*.

The data then appears in the language and with the regional settings specified in

- the user's Web browser options
- the run options
- the Cognos Connection preferences

Any text that users or authors add appears in the language in which they typed it.

Create a Multilingual Report in Report Studio

You can create a report in Report Studio, that can be viewed in different languages. For example, you can specify that text, such as the title, appears in German when the report is opened by a German user. You can also add translations for text objects, and create other language-dependent objects.

If you want the report to show data in different languages, the model must also be multilingual.

Steps

1. Create a report language variable.
2. In the work area, click the object that you want to modify based on one of the languages you selected.
3. In the **Properties** pane, double-click the **Style Variable** property.

   If you are changing the language of a text string, click **Text Source Variable** instead.
4. Click **Variable** and click the language variable you created.
5. In the **Values** box, select the languages you want the condition to support and click **OK**.

   Tip: A default value exists for the variable, and it is always selected.

6. Pause the pointer over the condition explorer button and click one of the possible languages for the variable.

   Tip: When you select a value in **Condition Explorer**, the Explorer bar becomes green to indicate that conditional formatting is turned on, and that any changes you make to the report applies only to the variable value.

7. In the **Properties** pane, specify the formatting that you want for the language.

   For example, to change the language of a text string, double-click the **Text** property, and select the new string.

8. Press Enter when you are done.

9. Repeat steps 7 to 9 for all other languages specified for the variable.

   Tip: When pausing the pointer over the condition explorer button, click (**No variable**) to view how the report looks when no variable is applied. Or you can triple-click the Explorer bar.

When you run the report, the report objects to which you applied the variable are formatted according to the browser's language.

Drill-through Access

Using drill-through access, you can move from one report to another within a session while maintaining your focus on the same piece of data. For example, you select a product in a sales report and move to an inventory report about that product.

Drill-through access helps you to build business intelligence applications that are bigger than a single report. Drill-through applications are a network of linked reports that users can navigate, retaining their context and focus, to explore and analyze information.

Drill-through access works by defining the target report using prompt parameters, and then using information from the source report to provide answers to those prompts. An important part of defining a drill-through path is defining which values to use for each prompt. If a prompt is answered by the drill-through value passed to it, the prompt page does not appear.

In Cognos 8, you can drill through

- between reports created in different packages against different data source types

  For example, you can drill through from an analysis against a cube to a detailed report against a relational data source.

- from one existing report to another report using Report Studio (p. 273)

- between reports in Report Studio, Query Studio, or Analysis Studio, exploring data in many ways

  For more information, see the user guide for the studio you are working in.

- from Cognos Series 7 to Cognos 8 (p. 653)

- from Metric Studio to other Cognos 8 reports by passing parameters using URLs

For more information, see the Metric Studio *User Guide*.

Understanding Drill-through Concepts

Before you set up drill-through access, you must understand the key concepts about drilling through. Knowing these concepts will help you to avoid errors so that report consumers drill through as efficiently as possible.

The target of drill-through access is always a saved report definition. The report can be created in Report Studio, Query Studio, or Analysis Studio.

Drill-through Paths

You can create a drill-through path in a source report in Report Studio, or using **Drill-through Definitions** in Cognos Connection. A drill-through path is the definition of the path that is taken when moving from one report to another, including how the data values are passed between the reports.

Using **Drill-through Definitions**, you can create a drill-through path from any report in the source package to any target report in any other package in Cognos Connection. This type of drill-through definition is stored in the source package and can be used when you want to drill between any combination of Analysis Studio, Query Studio, or Cognos Viewer reports in any package.

For any target report that contains parameters, the target parameters must be mapped to the correct metadata in the drill-through path. This ensures that the values from the source report are passed to the correct parameter values, and that the target report is filtered correctly.

A report-based drill-through path refers to a path created and stored in a Report Studio source report. The path is associated with a specific data column, chart, or cross tab, and is available only when users select that area of the report. The drill-through definition, if it exists, appears as a hyperlink in the source report when it is run.

The report-based drill-through is limited to Report Studio source reports and any target reports. This type of drill-through can be used when you want to pass the data item values or parameter results within a source report to the target report, the results of a report expression to a target report, or a URL link as a part of the drill-through definition.

Drilling Through to Different Report Formats

The settings in the drill-through definition determine how users see the report results. For example, the users may see the reports in Cognos Viewer as an HTML Web page, or the reports may open in Query Studio or Analysis Studio.

Reports can be output as HTML Web pages, as PDF, XML, CSV, or Excel formats. When you define a drill-through path, you can choose the output format. This can be useful if the expected use of the target report is something other than online viewing. If the report will be printed, output it as PDF; if it will be exported to Excel for further processing, output it as Excel or CSV, and so on.

If you define a drill-through path to a report that is created in Analysis Studio or Query Studio, the report can be run and opened in its studio instead of in Cognos Viewer. This can be useful if you

expect a consumer to use the drill-through target report as the start of an analysis or query session to find more information.

An example of a this type of drill-through action can be an application where a dashboard style report of high-level data can be drilled through to Analysis Studio to investigate items of interest. The Analysis Studio view can then be drilled through to a PDF report for printing.

Note: Drilling through to Report Studio is not supported for the Express Authoring Mode. The Report Studio Professional Mode does not display data results.

Bookmark References

When you drill through, the values that you pass are usually, but not always, used to filter the report using parameters. Cognos 8 Business Intelligence supports bookmarks within saved PDF reports so that a user can scroll a report to view the relevant part based on a URL parameter. For example, you may define a report with one page per product and add a bookmark of the product number on each page. Report consumers can then select a bookmark to see the product that they want.

When a bookmark in the source report is used in a drill-through definition, it provides the value for the URL parameter. When report consumers drill through using this definition, they see the relevant section of the target report.

Bookmark references are limited to previously run reports that are output as PDF and contain bookmark objects.

Members and Values

Dimensionally modeled data, whether stored in cubes or stored as Dimensionally Modeled Relational (DMR) data, organizes data into dimensions. These dimensions contain hierarchies. The hierarchies contain levels. And the levels contain members.

An example of a dimension is Locations. A Locations dimension may contain two hierarchies: Locations by Organization Structure and Locations by Geography. Either of these hierarchies may contain levels like Country and City.

Members are the instances in a level. For example, New York and London are members in the City level. A member may have multiple properties, such as Population, Latitude, and Longitude. Internally, a member is identified by a Member Unique Name (MUN) (p. 270). The method by which a MUN is derived depends on the cube vendor.

Relational data models are made up of data subjects, such as Employees, which are made up of data items, such as Name or Extension. These data items have values, such as Peter Smith.

In Cognos 8, the methods of drilling through available are:

- Dimensional (member) to Dimensional (member)
- Dimensional (member) to Relational (data item value)
- Relational (data item value) to Relational (data item value)

If the target parameter is a member, the source must be a member and must also be from a conformed dimension (p. 271).

If the target parameter is a value, the source can be either a value or a member. If the source is a dimensional member, you must ensure that the level or dimension is mapped to the target data item correctly in the drill-through definition. The business key from which the member is sourced must match the relational target value, which is most often the business key (p. 271).

Member Unique Names

The member unique name (MUN) is a unique identifier for a member in Cognos reports. It is stored in the report specification when the member is referenced in the report directly. The MUN is used in drill-through between OLAP data sources. The member keys in the MUN for the different OLAP data sources must match.

The MUN is used to find the member in the data source, which is similar to how business keys are used to find records in a table. For example, when you create OLAP dimension Products, you use the Product Line database column as s label for the members in your Product Line level. However, you use the Product Line Code business key from the database table to ensure that all the Product lines are unique in that level. The source value that you used to create the members is used in combination with the data source name, hierarchy, and level information in the member unique name.

If the MUN changes, members that are directly referenced in expressions, filters, or reports are no longer found. Changes to the MUN may be related to other changes. For example, changes to the hierarchy and level structures may change the level unique name, and changes to the business key values may change the member key path. Other factors that can affect the MUN are application changes during the design stage or over time, Cognos PowerCube category codes that are unpredictably unique, the production environment that has more members than the test environment, or removing the member from the data source.

To avoid potential problems, we recommend the following best practices when you build OLAP data sources:

- Use unique codes and keys within a dimension for the member keys.
- Define your OLAP and relational packages using unique conformed values for the source values (business keys) within similar dimensions or data values where drill-through between applications may be required.
- Ensure that the business keys and dimension metadata structure are the same in the production and test environments.
- Do not change the business keys in Framework Manager in the production environment.
- Resolve the non-unique keys in a dimension in the data source before you build the cube.

  Ensure that there are no duplicate source values in all levels of a dimension before you build a PowerCube. We do not recommend using the tilde character (~) in the category codes.

  For more information, see the section about uniqueness in the Cognos Series 7 *Step-by-Step Transformer*.

Conformed Dimensions

If you work with more than one dimensional data source, you may notice that some dimensions are structured the same, and some are not. The reason that dimensions can be structured differently is that the data sources may serve different purposes.

For example, a Customer dimension appears in a Revenue data store, but not in an Inventory data store. However, the Products dimension and the Time dimension appear in both data stores.

Dimensions that appear in multiple data stores are conformed if their structure is identical for all of the following:

- hierarchy names
- level names
- level order
- internal keys

Drilling through is possible between different dimensional data stores only if the dimensions are conformed, and if the dimension data store is of the same vendor type, such as Cognos PowerCube as the source and the target. In the previously mentioned example of the Revenue and Inventory data stores, it is possible to define the Products and Time dimensions differently for each data store. However, for drill-through between the Products and Time dimensions to work, their structures must be identical in each data store.

Business Keys

When performing a drill-through from a member to a relational value, the business key of the member is passed. This means that your relational target parameter must be set up using the data item with a matching value, which is most often the business key data item.

For example, employees are usually uniquely identified by an employee number, not by their name, because their name is not necessarily unique. When you drill through from a dimensional member to a relational data item, the value provided is the business key. Therefore, the parameter in the target report must be defined to accept a business key value. The exact logic used to define the business key value supplied depends on the cube vendor. For Cognos PowerCubes, the business key value is the **Source** property defined for the level in Cognos Transformer. Cognos Series 7 Transformer PowerCubes pass the source value if the drill-through flag was enabled before the cube was built. Otherwise, the category code is used.

In Report Studio, you can determine what the member business key is using an expression such as `roleValue('_businessKey',[Camping Equipment])`. This expression is case sensitive.

MSAS 2005 multi-part business keys are not supported in drill-through operations.

Tip: When other users run your drill-through report, you may not want them to be prompted for a business key. In Report Studio, you can build a prompt page with a text that is familiar to the users, but filters on the business key. Your Framework Manager modeler can also set the **Display Item Reference** option for the **Prompt Info** property to use the business key when the data item is used in a prompt.

Drilling Through on Dates Between PowerCubes and Relational Packages

Usually, drilling through from OLAP to relational packages requires that the target report parameter is set using the business key in the relational data. However, this method does not work well for dates. OLAP data sources typically view dates as members, such as Quarter 1 2006, while relational data sources view dates as ranges, such as 1/Jan/2006 to 31/March/2006.

A special feature exists for drilling through between PowerCubes and relational packages. Ensure that the target report parameter is set up using `in_range`. Here is an example:

```
[gosales_goretailers].[Orders].[Order date] in_range
?Date?
```

Also ensure that the drill-through definition maps the parameter at the dimension level and that the PowerCube date level is not set to suppress blank categories. Enabling the option to suppress blank categories in the Transformer model before you build the cube may cause the drill-through on dates to be unsuccessful. This happens because there are missing values in the range.

Drilling Through Between Packages

You can set up drill-through access between different packages. The two packages can be based on different types of data source, but there are some limits.

The following table shows the data source mappings that support drill-through access.

Source data source	Target data source
OLAP	OLAP **Note:** OLAP to OLAP drill through is supported only if the data source type is the same, for example, MSAS to MSAS.
OLAP	Dimensionally modeled relational
OLAP	Relational data **Note:** For more information, see "Business Keys" (p. 271).
Dimensionally modeled relational	Dimensionally modeled relational
Dimensionally modeled relational	Relational
Relational	Relational

Scope

Scope is specific to drill-through definitions created using **Drill-through Definitions** in Cognos Connection. It defines when the target report is shown to the users, based on the items they have in the source report.

Usually, you define the scope of a drill-through path to match a parameter that it passes. For example, for a target report that contains a list of employees, typically you only want to display the report as an available drill through-choice when a user is viewing employee names in a source report. If employee names are not in the source report and the scope was set on the employee name in the drill-through definition, the employee report is suppressed from the list of available drill-through target reports in the **Go To** page.

In report-based drill-through access, where the drill-through path is associated with a specific report column, the column serves as the scope.

Set Up Drill-through Access in a Report

Set up drill-through access in a source report to link two reports containing related information. You can then access related or more detailed information in one report (the target) by selecting one or more data item values from another report (the source). In addition, you can pass parameter values from the source report to the target report. This means that you can filter the target report using the same prompted filter values in the source report.

You can also drill through within the same report by creating bookmarks (p. 141), and create drill-through definitions in the package. Users can use package drill-through definitions to navigate to a target report from an Analysis Studio analysis, a Query Studio report, or a Report Studio report. For more information, see the *Administration and Security Guide*.

If you have the Cognos 8 software development kit (SDK) (p. 78), you can use URLs to set up drill-through access to and from third-party sources.

You can also drill through from a map (p. 128).

If you are using an SAP BW data source for the target report, and if the target report contains a variable for a hierarchy node, values from the source report can be values only of the data item representing the leaf-level identifier of the hierarchy.

Before you begin, ensure that you have a report that will serve as the source report and another report that will serve as the target report.

Steps

1. Open the target report.
2. Create a parameter that will serve as the drill-through column or that will be used to filter the report.

   For example, to drill through or filter Product line, create a parameter that looks like this:

   [Product line]=?prodline_p?

   Tip: Use the operators `in` or `in_range` if you want the target report to accept multiple values or a range of values.
3. In the **Usage** box, specify what to do when the target parameter is not fulfilled as part of a drill-through:
   - To specify that users must click a value in the source report, click **Required**.

If the parameter is not fulfilled and it is required, users are prompted to choose a value.

Note: Drill-through definitions that have been authored inside a Report Studio report appear to the consumer as a blue hyperlink. Clicking the hyperlink is how they drill through to the target report and select the value they want passed to the target.

A consumer can choose to select multiple items within a single column and when the drill-through target report is run, the values from each row selected will be passed to the target parameters. This occurs as an *and* condition.

- To specify that users do not need to click a value in the source report, click **Optional**.

  Users are not prompted to choose a value if the parameter is not fulfilled. The value will be unfiltered.

- To specify not to use the parameter, click **Disabled**.

  The parameter is not used during the drill-through.

4. To create other parameters, repeat steps 2 to 3.
5. Open the source report.
6. Click the element in the report that will serve as the drill-through object.

   You can select a data item that your report users are likely to choose to drill on for more detailed information, such as an Employee Name data item. When defining the drill-through path in Report Studio, you can choose to pass a value from a different data item hidden from the user for display purposes, but still in the query. For example, your users see the Product Name data item and can drill through on that item but the drill-through definition passes the Product Number value for that particular product name that the user has chosen.

   Tip: If you are passing only parameter values to the target report, you do not have to drill on a data item. Instead, you can drill on any object in the report, such as the report title. This means that you can drill from outside the context of a query.

7. Click the drill-through definitions button or from the **Properties** pane, double-click the **Drill-Through Definitions** property.
8. Click the new drill-through definition button.

   A drill-through definition is created.

   Tip: To change the drill-through name, click the rename button, type the new name, and click **OK**.

9. On the **Target report** tab, click the ellipsis (...) button next to the **Report** box, and select the drill-through target report.
10. Under the **Parameters** box, click the edit button .

    Each required and optional parameter defined in the target report appears in the **Parameters** dialog box.

11. For each parameter, click **Method** and do one of the following:

- To specify not to pass any value, click (**Default**).

  That is, the default method is to not pass any value.

- To specify not to use this parameter, click **Do not use parameter**.

  The target report is not be filtered by this parameter.

- To pass values from a data item, click **Pass data item value**, click **Value**, and then click the data item that you want.

  Values for the selected data item are passed to the target report.

- To pass values from a source report parameter, click **Pass parameter value**, click **Value**, and then click the parameter that you want.

  Values for the selected parameter are passed to the target report.

12. Click **OK**.
13. In the **Action** box, decide how the target report will be viewed when users click the drill-through column in the parent report:
    - To run the target report, click **Run the report**.

      Run the target report when you want to see the latest data.

    - To view the most recently saved output version of the target report, click **View the most recent report**.

      Saved output versions are created in Cognos Connection.

    - To use the default action specified for the report in Cognos Connection, click (**Default**).
14. If you chose to run the target report in the previous step, in the **Format** box, click the output format you want for your report.

    Tip: Click (**Default**) to run the report using the default format specified for the report in Cognos Connection.
15. If you want the target report to appear in a new window, select the **Open in new window** check box.
16. In the **Display prompt pages** box, decide whether to display prompt pages:
    - To always display prompt pages when the drill-through action occurs, regardless of whether values were specified for the required parameter values, click **Always**.
    - To use the prompt settings of the target report (specified by the **Prompt for Values** checkbox in Cognos Connection - Report Properties - Report tab) to determine whether to show the prompt pages, click **Based on the default prompt settings of the target report**.
    - To not display prompt pages when the required parameter values are provided by the drill-through definition, click **Only when required parameter values are missing**. This is the default.

The drill-through object appears as a blue hyperlink in the report. This drill-through action can also be started by clicking the **Go To** button . When users run the source report, they can click the object to drill through to the target report. If more than one target report is available, the **Go To** page appears with the list of available targets.

Tip: If you run the target report directly, you are prompted to select a value for the report to run on.

Specify the Drill-through Text

You can specify the drill-through text that appears when users can drill through to more than one target. For example, if users from different regions view the report, you can show text in a different language for each region.

Steps

1. Right-click the drill-through object and click **Drill-Through Definitions**.
2. If more than one drill-through exists for the object, in the **Drill-Through Definitions** box, click a drill-through definition.
3. Click the **Label** tab.
4. To link the label to a condition, in the **Condition** box, do the following:
   - Click **Variable**, and click an existing variable or create a new one.
   - Click **Values**, and click one of the variable's possible values.
5. In the **Source type** drop-down list, click the source type you want to use.
6. If the source type is **Text**, click the ellipsis (...) button beside the **Text** box and type the text you want.
7. If the source type is **Data Item Value** or **Data Item Label**, click **Data Item**, and click a data item.
8. If the source type is **Report Expression**, click the ellipsis (...) button beside the **Report Expression** box and define the expression.
9. If the label is linked to a condition, repeat steps 5 to 8 for the remaining possible values.
10. Click **OK**.

When users run the source report and click a drill-through link, the **Go to** page appears. The drill-through text you specified appears for each target. If you did not specify the drill-through text for a target, the drill-through name is used.

Example - Create a Drill-through Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales by order method for each product line, and allows users to drill through from the sales report to view the product details for any item selected.

You will create two reports, one that contains the details for the item, and another that lists the product sales for the order methods selected by users.

Steps

1. In the Cognos Connection **Welcome** page, click the **Launch** link, and then click **Report Studio**.
2. Select the package **GO Data Warehouse (query)**.
3. In the **Welcome** dialog box, click **Create a new report or template.**
4. In the **New** dialog box, click **List** and click **OK.**
5. In the **Insertable Objects** pane, on the **Source** tab , add the following data items to the list by double-clicking them:
   - **Order method**
   - **Product line**
   - **Product type**
   - **Product name**
   - **Introduction date**
   - **Product image**

   Tip: You can find these data items in the **Order method** and **Product** folders.
6. Group the **Order method**, **Product line**, and **Product type** columns.
7. Click the filters button .
8. Click the add button, and type the following in the **Expression Definition** box:

   [Product name]=?p_PN?
9. Click **OK.**
10. Repeat steps 8 to 9 to create the following filter:

    [Order method] in ?p_OM?
11. Change the title of the report to **Product Details.**
12. Save the report as **Product Details.**
13. From the **File** menu, click **New** to create a new report.
14. In the **New** dialog box, click **List** and click **OK.**
15. In the **Insertable Objects** pane, on the **Source** tab, add the following data items to the list by double-clicking them:
    - **Order method**
    - **Product line**

- **Product type**
- **Product name**
- **Quantity**
- **Revenue**

Tip: You can find these data items in the **Sales fact** and **Product** folders.

16. Group the **Order method**, **Product line**, and **Product type** columns.
17. Click the filters button .
18. Click the add button, and type the following in the **Expression Definition** box:

    [Order method] in ?p_OM?
19. Click **OK**.
20. Right-click the **Product Name** column and click **Drill-Through Definitions**.
21. Click the new drill through button .
22. Under **Report,** click the ellipsis (...) button and select the **Product Details** report you created previously.
23. Click **Action** and click **Run the report**.
24. Click **Format** and click **HTML**.
25. Click the edit button .
26. For the item **p_OM**, under **Method,** select **Pass parameter value**, and select **p_OM** for the **Value**.
27. For the item **p_PN**, under **Method,** select **Pass data item value,** and select **Product name** for the **Value**.
28. Click **OK** twice.
29. Change the title of the report to **Product Revenue by Order Method**.
30. Save the report as **Product Revenue by Order Method**.
31. Click the run report button .

When the report runs, you are prompted to select one or more order methods. When you click **OK**, the list shows the product names as clickable links. When you click a product name, the second report runs, showing the order method that was selected in the prompt page of the source report and the product (the source report's column value) selected in the first report.

Create a Drill-up/Drill-down Report

You can create a report that allows the reader to drill down to lower level data sets or drill up to higher-level data sets, provided you are using a dimensional or dimensionally modeled relational (DMR) data source. This will allow you to rerun the report for a different data set without having to regenerate the report.

An example of a dimensional hierarchy might be:

Years - Year - Quarter - Month

Drilling up and down allows you to view more general or more detailed information on your data within such a predefined dimensional hierarchy, without having to create different reports.

Before you begin, ensure that you are using a dimensional data source.

Steps

1. From the **Data** menu, click **Drill Behavior**.
2. On the **Basic** tab, in the **Report drill capabilities** box, select the **Allow drill-up and drill-down** check box.

   By default, Report Studio determines which items can be drilled on, based on the dimensional structure.

   Tip: If you want the report to be used as the source during a package drill-through, select the **Allow package based drill-through** check box and click **OK**. For more information about package drill-through, see the *Administration and Security Guide*.
3. To disable drill-up or drill-down for a data item, select the data item in either the **Disable drill-up for** box or the **Disable drill-down for** box.
4. From the **Advanced** tab, you can change the drill-up or drill-down behavior for any data item by selecting the data item and then choosing one of the following behaviors.

Behavior name	Drill-up behavior	Drill-down behavior
Preserve	The data item's value will remain unchanged.	The data item's value will remain unchanged.
Empty Set	The set of values associated with this data item is set to be the empty set (novalues). For crosstabs, the data item will effectively be removed from the report.	The data item will be removed from the report.

Behavior name	Drill-up behavior	Drill-down behavior
Replace Item	The data item's value will change to become the parent of the item drilled on (if a lowest-level summary was drilled up on), or the grandparent (if a lowest-level detail of a dimension was drilled up on).	The data item's value will change to become the item drilled on.
Replace Expression	The data item's value will change to become the children of the parent of the item drilled on (if a lowest-level summary was drilled up on), or the children of the grandparent (if a lowest-level detail of a dimension was drilled up on).	The data item's value will change to become the children of the item drilled on.
Ancestor	The data item's value will change to become the data value of the ancestor "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.	The data item's value will change to become the data value of the ancestor "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.
Depth Based Expression	The data item's value will change to become all data items "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.	The data item's value will change to become all data items "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.
Change Expression	The data item's value will change to become the value of some other data item in the query.	The data item's value will change to become the value of some other data item in the query.

5. Click **OK**.

The report will generate links for any item that can be drilled up or down on.

You can perform drill-down or drill-up by right-clicking the data item and choosing the action from the context menu. The menu items will be disabled if an item cannot be drilled up or down on.

Member Sets

Member sets are used to group data items that are logically related for various actions, such as drill actions, zero suppression, and ranking. They can be defined as a flat list or, as a tree structure, where member sets that are in the same parent chain are considered related.

For example, for drill operations, a member set defines the set of items that can potentially change when a given item in the set is drilled on. The values of other items in the query or even those in the same hierarchy are preserved when any item in this set is drilled on. Usually, a member set references items that have logical roles in a drill action, such as a detail, a summary of the detail, or a parent of the detail. A single data item can belong to only one member set.

If you do not define member sets for items, the Cognos 8 server associates items into default member sets and behaviors using simple dimension rules on item expressions. You can override the behavior for a particular item while other items continue to use the default.

When you define a member set, you must explicitly define behaviors for each item in the set. Items in the set that have no behaviors have their values preserved.

Drill behaviors always act from a root member set. This means that when an item is drilled on, the root member of its member set is found and all items from the root down are processed. Although calculations and aggregates are not directly related by hierarchy, they respond because of their dependence on the items upon which they are based.

Create a Member Set

Create a member set when you want to define a non-default drill behavior. You specify what items respond to a drill action by adding them to the member set.

Steps

1. Pause the pointer over the query explorer button and click the query in which you want to create a member set.
2. In the **Properties** pane, click the **Define Member Sets** property and click **Yes**.
3. Click the **Member Sets** tab.
4. In the **Insertable Objects** pane, drag the items that you want to the work area.

   To define a member set as a tree structure, drag the item that will serve as the root item of the set to the work area, and then drag other items over the root item to create the tree structure.

   You can also nest member sets.

Create a Master Detail Relationship

Create a master detail relationship to deliver information that would otherwise require two or more reports. For example, you can combine a list with a chart. The list can contain product lines and the chart can show details for each product line.

Master detail relationships must appear in nested frames to produce the correct results. You can create a master detail relationship in two ways:

- Use a parent frame for the master query and a nested frame for the detail query.
- Associate a report page with the master query and use a data container, such as a list or crosstab, on the page for the detail query.

Note that you cannot display parent data in the child frame or child data in the parent frame. Also, you cannot perform calculations across master detail queries.

You can use a master detail relationship to show data from separate data sources in a single report. However, the data sources must be contained in the same package.

If you are working with an SAP BW data source, you cannot use a data item from the master query that contains non-ASCII values.

To create a master detail relationship using queries that reference two different dimensional data sources, you must first link the members by aliasing the levels that contain the members.

Steps

1. To use a parent frame for the master query and a nested frame for the detail query, do the following:
   - In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **List**, **Repeater Table**, or **Repeater** to the report.
   - Add a second data container to the object you previously inserted.

     You can insert a list, crosstab, chart, repeater table, or repeater into a list. You can add a list to a repeater table or repeater.
   - Add the data items you want to both data containers.
2. To associate a report page with the master query and use a data container on the page for the detail query, do the following:
   - Click anywhere in the report page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.
   - Click the **Query** property and then choose a query from the drop-down list.
   - In the **Insertable Objects** pane, on the **Toolbox** tab, drag a data container to the report.
   - Click **OK** to create the detail query.

3. To link a data item in the master query to a parameter in the detail query instead of to another data item, create the parameter in the detail query.

   Use parameters when you want to filter values at a lower level in the detail query.

   - Pause the pointer over the query explorer button and click the detail query.
   - In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Filter** object to the **Detail Filters** box.
   - In the **Expression Definition** box, create the parameter (p. 205).
   - Click **OK**.

4. Pause the pointer over the page explorer button , and click the report page.
   - Click anywhere in the report page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.

5. Click the data container containing the details.
6. From the **Data** menu, click **Master Detail Relationships**.
7. Click the **New Link** button.
8. In the **Master** box, click the data item that will provide the primary information.
9. To link the master query to a data item, in the **Detail** box, click the data item that will provide the detailed information.
10. To link the master query to a parameter, in the **Parameters** box, click the parameter that will provide the detailed information.
11. Repeat steps 7 to 10 to create other links.

    Tip: To delete a link, click it and press the Delete key.

12. Click **OK**.

Tip: To avoid seeing the same data item twice in the report, click the data item in the data container driven by the detail query, and click the cut button. This removes the item from the report display, but keeps it in the query.

Link Members from Two Dimensional Data Sources

If you create a master detail relationship using queries that reference two different dimensional data sources, you may need to create a relationship between levels with the same member captions but different Member Unique Names (MUNs). For more information about MUNs, see "Recommendation - Use Member Unique Name (MUN) Aliases" (p. 360).

For example, you may want to link the Americas member in the Sales territory level in the Great Outdoors cube with Americas in the GO Subsidiary level in the Great Outdoors Finance cube. To create the relationship, you must alias the levels that contain the members that you want to link.

Steps

1. In the parent frame or the report page, double-click the level that contains the member that will provide the primary information.

   The **Data Item Expression** dialog box appears.

2. In the **Expression Definition** box, use the expression in the following function:
   `caption(`*`expression`*`)`

   For example, `caption([great_outdoors_company].[Sales Territory].[Sales Territory].[Sales territory])`

3. Click **OK**.

4. Repeat steps 1 to 3 for the level in the data container that contains the details.

   A caption alias is created for each level. You can now use the caption alias for each level to create a master detail relationship using member captions, which are the same, instead of MUNs, which are different.

You can now create the master detail relationship between the two levels.

Chapter 7: Try It Yourself Exercises

If you have some experience with Report Studio and want to improve your skills in report writing, this chapter is for you. Each topic gives you some guidelines on how to create each sample report. If you need help, links to more detailed instructions are available.

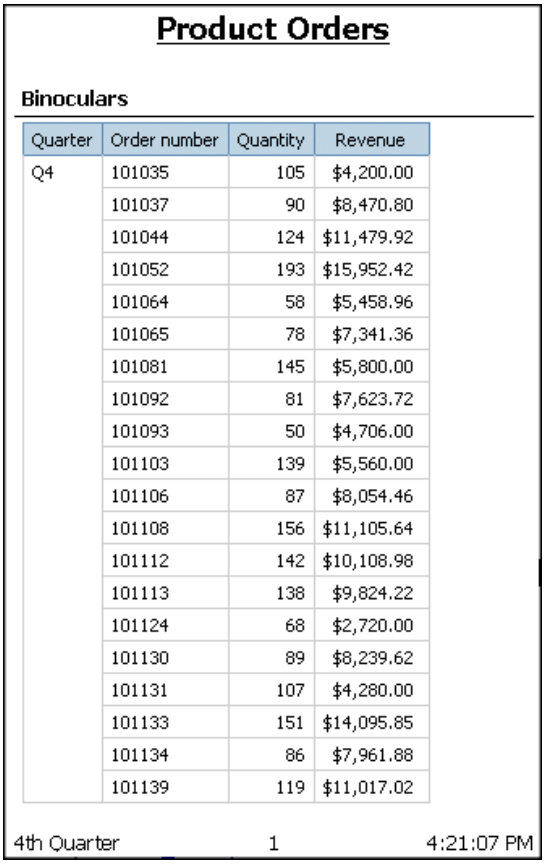
If you have not used Report Studio before, see the Report Studio Tour first to learn basic skills.

Try It Yourself - Create a List Report

Use list reports to show detailed information from your database, such as product lists and customer lists.

In this topic, you learn how to create a list report that shows revenue for each product for the last quarter of the current year.

It should take 15 to 20 minutes to complete this topic, and your report will look like this.



Product Orders

Binoculars

Quarter	Order number	Quantity	Revenue
Q4	101035	105	$4,200.00
	101037	90	$8,470.80
	101044	124	$11,479.92
	101052	193	$15,952.42
	101064	58	$5,458.96
	101065	78	$7,341.36
	101081	145	$5,800.00
	101092	81	$7,623.72
	101093	50	$4,706.00
	101103	139	$5,560.00
	101106	87	$8,054.46
	101108	156	$11,105.64
	101112	142	$10,108.98
	101113	138	$9,824.22
	101124	68	$2,720.00
	101130	89	$8,239.62
	101131	107	$4,280.00
	101133	151	$14,095.85
	101134	86	$7,961.88
	101139	119	$11,017.02

4th Quarter 1 4:21:07 PM

Steps

1. Create a new list report that uses the sample package named **GO Data Warehouse (query)** .

2. Add the following data items to the report:
   - **Product type** (in **Sales and Marketing (query)**, **Sales (query)**, **Product**)
   - **Quarter** (in **Time Dimension**)
   - **Order number** (in **Sales orders**)
   - **Product name** (in **Product**)
   - **Quantity** (in **Sales fact**)

   Tip: Use the **Source** tab in the **Insertable Objects** pane.
3. Create this query calculation named **Revenue**:

   [Sales (query)].[Sales fact].[Unit cost]*[Quantity]

   Tip: Use the **Toolbox** tab in the **Insertable Objects** pane.
4. Group the **Product type** column to make the report easier to read. Then group the **Quarter** column.
5. Make the **Product type** column appear as a section header in the report.
6. Add a total to the **Revenue** column to view total revenue for each product type.
7. Create this tabular filter to view revenue for the fourth quarter.

   [Quarter]=('Q4')
8. Change the **Data Format** property for the **Revenue** column to currency .
9. Change the title text placeholder to this text:

   Product Orders
10. Add the following text to the bottom cell, replacing the date object:

    4th Quarter
11. Run the report to view what it will look like for your users.

Need More Help?

- Add Data to a Report
- Add a Summary
- Create a Calculation
- Filter Data
- Group Data
- Run a Report

Try It Yourself - Create a Crosstab

Use crosstab reports to compare information that uses one or more criteria. The values at the intersection points of rows and columns show summarized information.

In this topic, you learn how to create a crosstab that shows the cost of goods sold for each product line and product type by quarter.

It should take 15 to 20 minutes to complete this topic, and your report will look like this.

Profitability by Product Line

		2007			2006	
		Gross profit	Revenue	*Cost of Goods Sold*	Gross profit	Reven
Personal Accessories	Glacier Deluxe	1,741,826.22	3,928,757.22	*2,186,931*	1,669,390.15	3,958
	Glacier GPS Extreme	5,639,265.95	12,034,715.22	*6,395,449.27*	5,921,887.95	12,249
	TX	8,306,303.78	18,172,476.5	*9,866,172.72*	14,701,866.32	32,35
	Pocket Gizmo	2,675,373.35	4,380,582	*1,705,208.65*	3,454,153.34	5,68
	Seeker 35	2,235,849.84	7,754,925.78	*5,519,075.94*	2,574,835.2	8,93
	Zone	9,434,315.06	28,021,707.95	*18,587,392.89*	16,263,229.54	49,62
	Mountain Man Deluxe	2,269,799.48	4,563,194.48	*2,293,395*	2,493,495.99	4,972
	Polar Extreme	347,639.66	701,512.16	*353,872.5*	499,545.53	998
	Ranger Vision	5,622,674.4	10,525,086.6	*4,902,412.2*	5,891,875.71	11,71

Steps

1. Create a new crosstab that uses the sample package named **GO Data Warehouse (query)** .
2. Add the following data items to the report:
   - **Date** (in **Sales and Marketing (query)**, **Sales (query)**, **Time dimension**) in the columns
   - **Product line** (in **Product**) in the rows
   - **Revenue** (in **Sales fact**) as the first measure
   - **Gross profit** (in **Sales fact**) as the second measure

   Tip: Use the **Source** tab in the **Insertable Objects** pane.
3. Nest **Product type** (in **Product**) after **Product line** in the rows.
4. Create this query calculation named **Cost of Goods Sold** and insert it in the columns:

   [Revenue]-[Gross Profit]

   Tip: Use the **Toolbox** tab in the **Insertable Objects** pane.

5. Change the **Data Format** property for **Revenue, Gross profit,** and **Cost of Goods Sold** to currency.
6. Change the placeholder title to this text:

   Profitability by Product Line
7. Run the report to view what it will look like for your users.

Need More Help?

- Add Data to a Report
- Create a Calculation
- Create a Nested Crosstab Report
- Crosstab Reports
- Run a Report

Try It Yourself - Create Charts

Use charts to reveal trends and relationships that are not evident in tabular reports.

Report Studio gives you flexibility in how you organize data in charts.

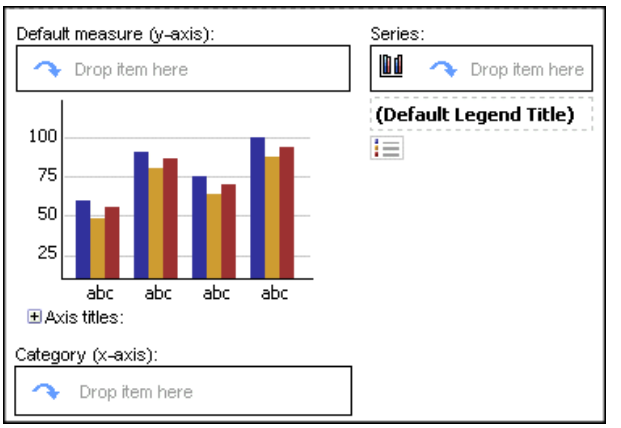
In this topic, you learn how to create charts that show different information.

It should take 15 to 20 minutes to complete this topic.

Let's take a look at the basic chart interface.

Steps to Create a Basic Column Chart

1. Open Report Studio with the **GO Data Warehouse (query)** package.
2. Create a new **Chart** report.
3. Click **OK** to open the default column chart.



You can click on the white space around the chart to select the whole chart, or you can click on an individual item, such as an axis icon or chart type icon, to find a given chart property.

The **Series** drop zone defines the items that appear in the legend. This is the legend edge. The **Category** (**x-axis**) drop zone defines the ordinal edge.

Let's chart some Great Outdoors Company data.

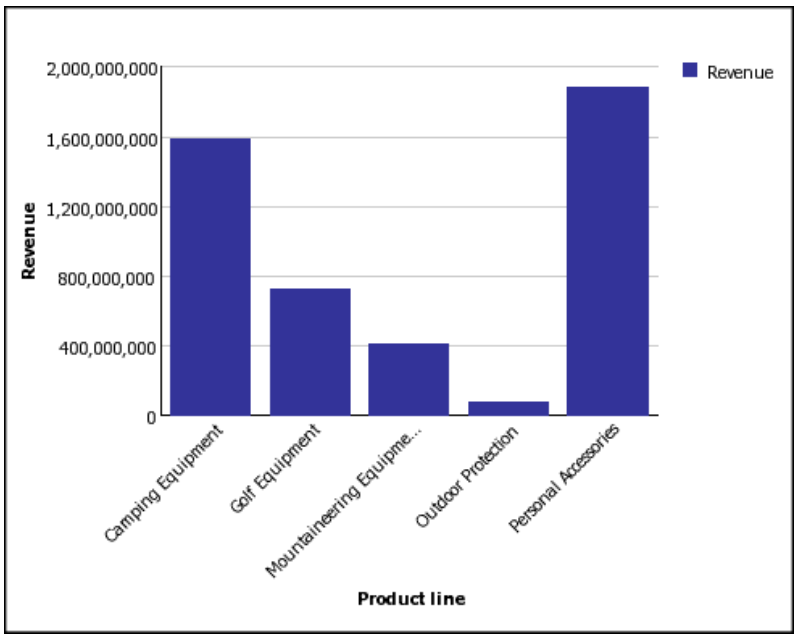
Steps to Add Data to a Chart

1. Drag the following items to the chart:
   - **Revenue** (in **Sales fact**) to the **Default measure** (**y-axis**) drop zone
   - **Product line** (in **Product**) to the **Category** (**x-axis**) drop zone

   Tip: Use the **Source** tab in the **Insertable Objects** pane.

   Column, bar, line, area, and clustered charts are based on the same combination chart for flexibility in charting.

2. Run the report to see your chart.



3. Return to Report Studio.

We want to show the revenue value on each column in the chart.

Steps to Show Values in a Chart

1. Click the bar icon in the **Series** drop zone.
2. In the **Properties** pane, under **Chart Labels**, change the **Values** property to **Show**.
3. Run the report to see your chart.

   You can customize the font properties for the values shown in the chart, independent of the axis labels.

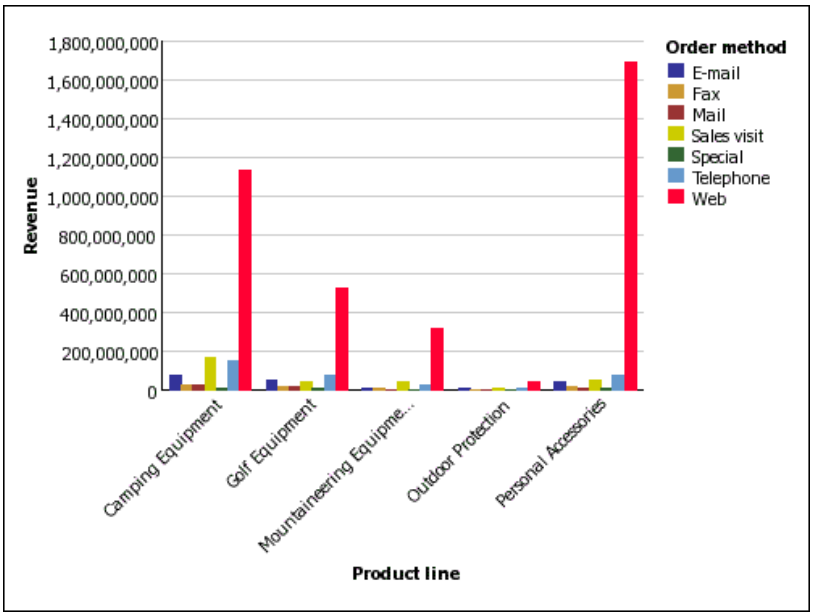
4. From the **Structure** menu, clear **Lock Page Objects** to unlock the chart objects.
5. Select the chart body by clicking between the axis.
6. In the **Properties** pane, increase the font size and change the foreground color to red.
7. Run the report to see your chart.

   This makes the values easier to see. After we add more data, showing the values clutters the chart, so we will remove them from view again.
8. Return to Report Studio.
9. Click the bar icon in the **Series** drop zone.
10. In the **Properties** pane, under **Chart Labels**, change the **Values** property to **Hide**.

Let's group this chart based on **Order Method**.

Steps to Group Data in a Chart

1. Drag **Order method** (in **Order method**) to the **Series** drop zone.
2. Run the report to see your chart.



3. Return to Report Studio.

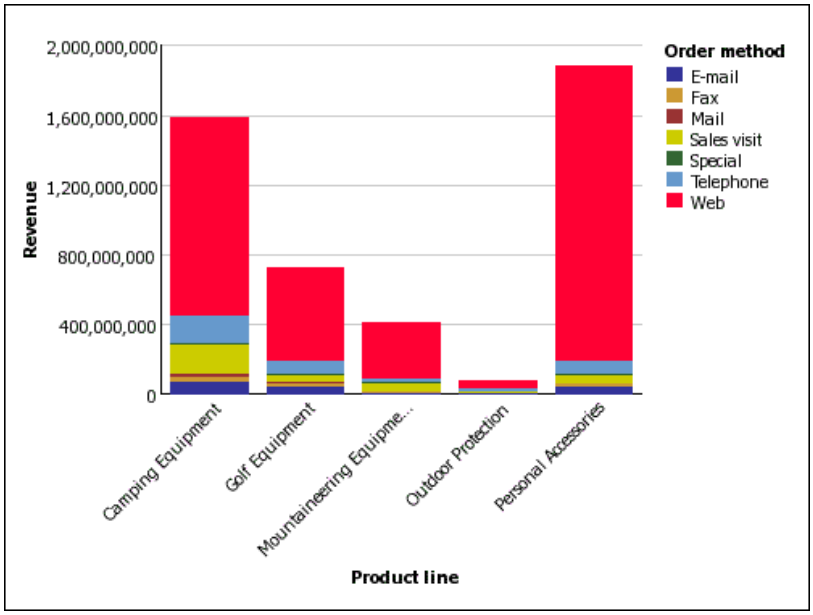
Let's see what other kind of charts we can make with this data.

Steps to Change the Chart Configuration

1. Click the **Order method** legend item icon .

   There are many properties to let you customize how the columns are shown.
2. In the **Properties** pane, set the **Grouping Type** property to **Stacked**.

3. Run the report to see your chart.



4. Return to Report Studio.

You can also use the properties in the **Properties** pane to show the data as a 100% stacked column chart, show or hide borders, and so on.

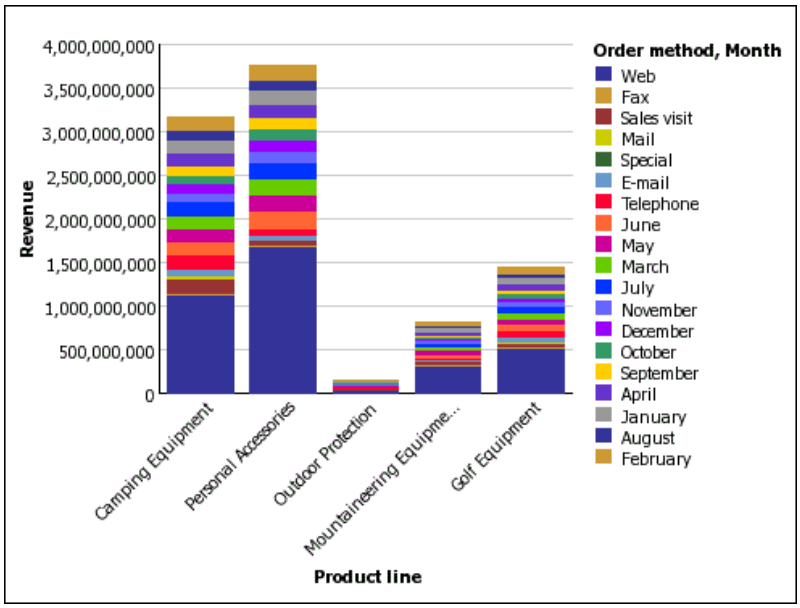
Let's add some more data to this chart.

Steps to Combine Sets of Data in Charts

1. Add **Month** (in **Time dimension**) to the **Drop item here** area in the **Series** drop zone.

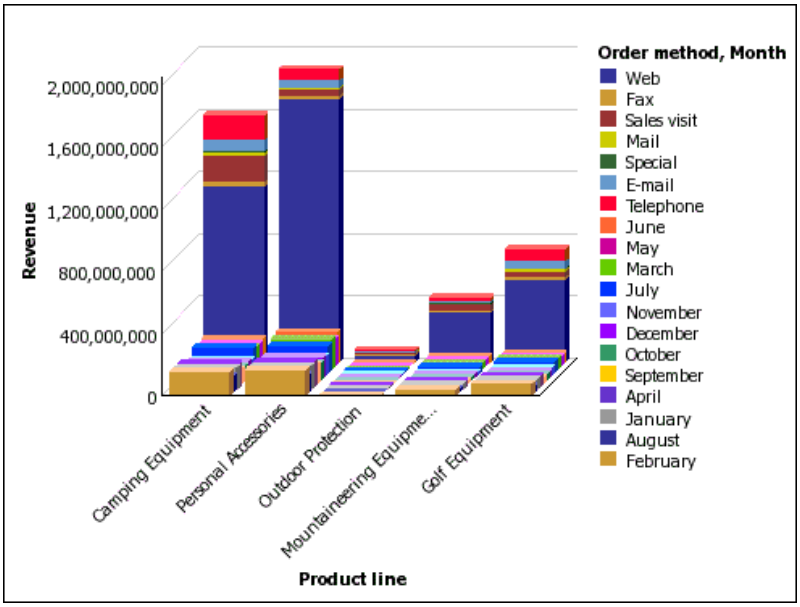
   This is how you union together the **Order method** and **Month** items on the legend edge.

2. Run the report to see your chart.



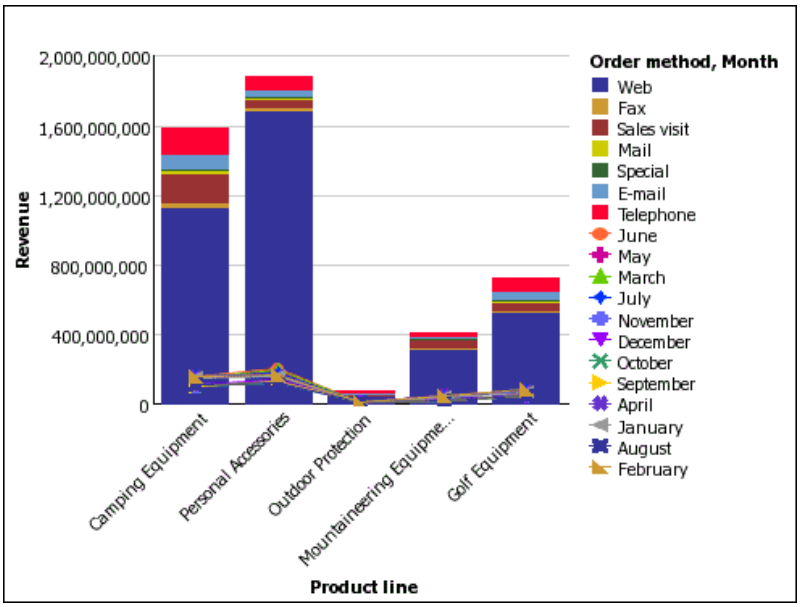
Report Studio places another set of data values on the stack.

3. Return to Report Studio.
4. Click the **Month** legend item icon.
5. In the **Properties** pane, set the **Grouping Type** property to **Clustered**.
6. Run the report to see your chart.



You can combine stacked and unstacked charts.

7. Return to Report Studio.
8. Click the **Month** legend item icon.
9. Set the **Chart Type** property to **Line**.
10. Run the report to see your chart.



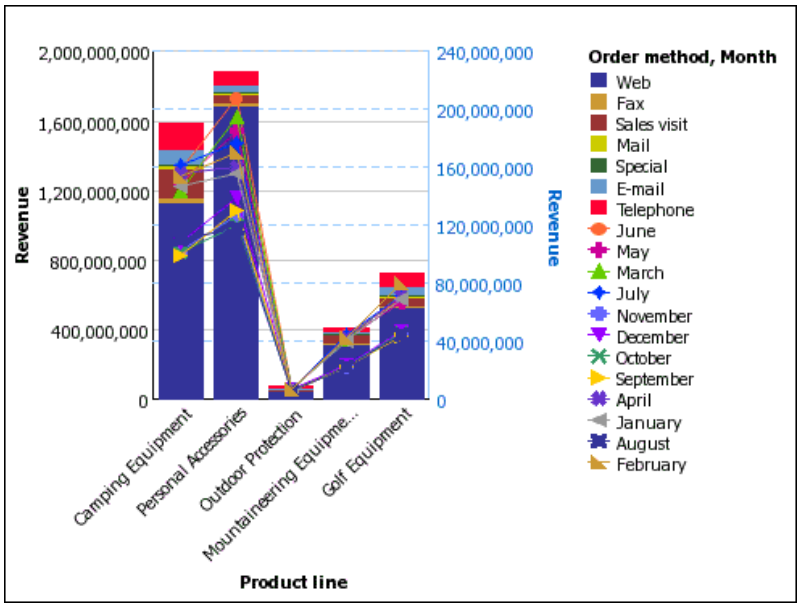
You can combine line and stacked charts. Let's improve the presentation of the **Month** values by showing them on their own y-axis.

Steps to Change the Chart Presentation

1. Return to Report Studio.
2. Click the **Month** legend item icon.
3. Set the **Axis Assignment** property to **Y2 Axis**.

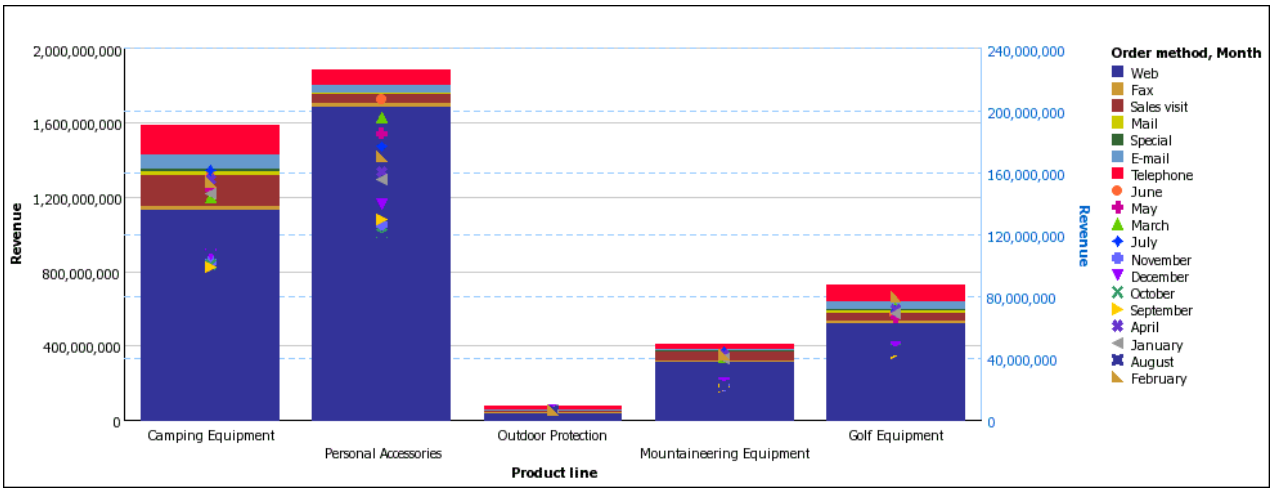
   The chart image now shows two y-axes. There are two ways to view this second y-axis.
4. Select the chart background, and set the **Y2 Axis Position** property to **Bipolar**.
5. Run the report to see your chart.

   The line chart appears under the stacked column chart.
6. Return to Report Studio.
7. Select the chart background, and set the **Y2 Axis Position** property to **Dual**.
8. Run the report to see your chart.



   Now the chart is too narrow. We also prefer points instead of lines.
9. Return to Report Studio.
10. Click the **Month** legend item icon and, in the **Properties** pane, change the **Line** property from **Yes** to **No**.

    You can also use the **Properties** pane to resize the points, restyle the lines, and so on.
11. Select the chart background, set the **Size & Overflow** property **Width** to 1000 px, and click **OK**.
12. Run the report to see your chart.



13. Return to Report Studio.

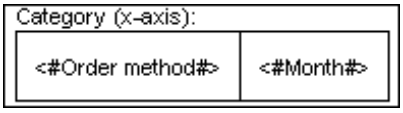
    There is another way to show the second axis.

14. Set the **Y2 Axis Position** property to **Bipolar**.

    You can pivot a chart like a crosstab.

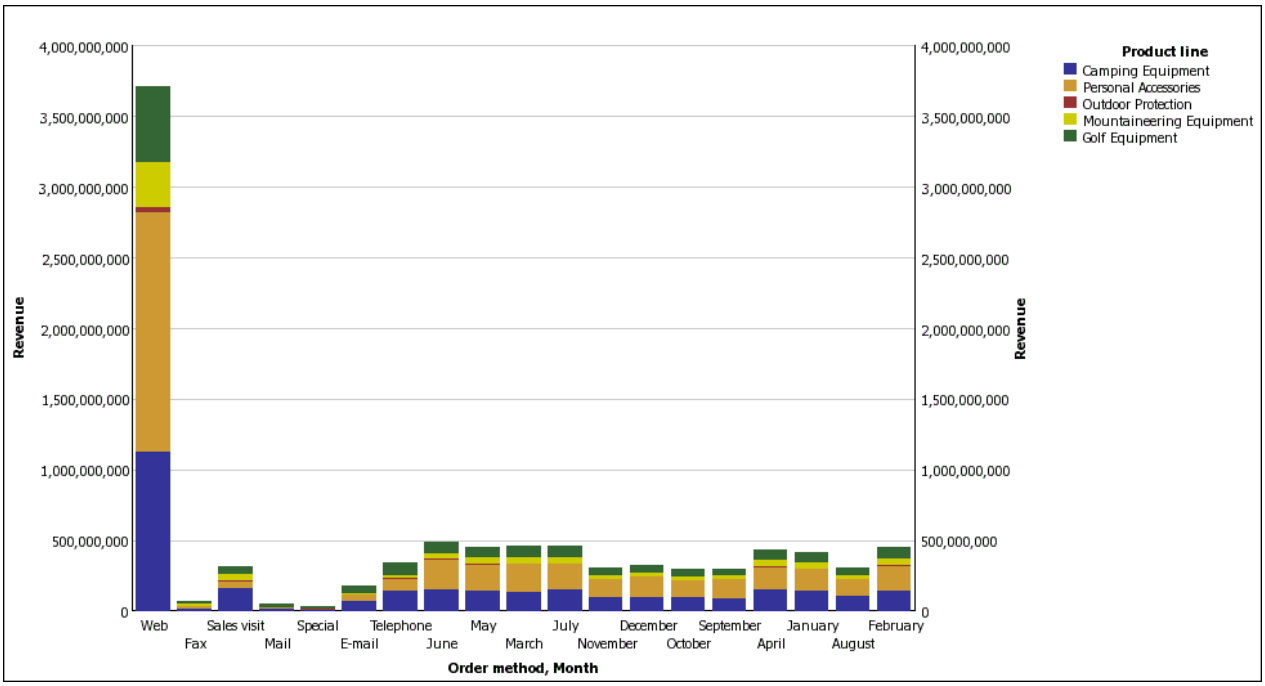
15. Click the swap rows and columns button on the toolbar.

    There are now two ordinal edge entries: **Order method** and **Month**.



    You can union and nest data items on both edges equally. There is an additional drop zone on the legend edge because multiple legend entries are more common in charting than multiple ordinal edge entries.

16. Run the report to see your chart.



17. Return to Report Studio.

    You can view this as a horizontal bar chart.

18. Select the chart and set the **Chart Orientation** property to **Horizontal**.

19. In the **Size & Overflow** property, delete the **Width** value.

20. Run the report to see your chart.

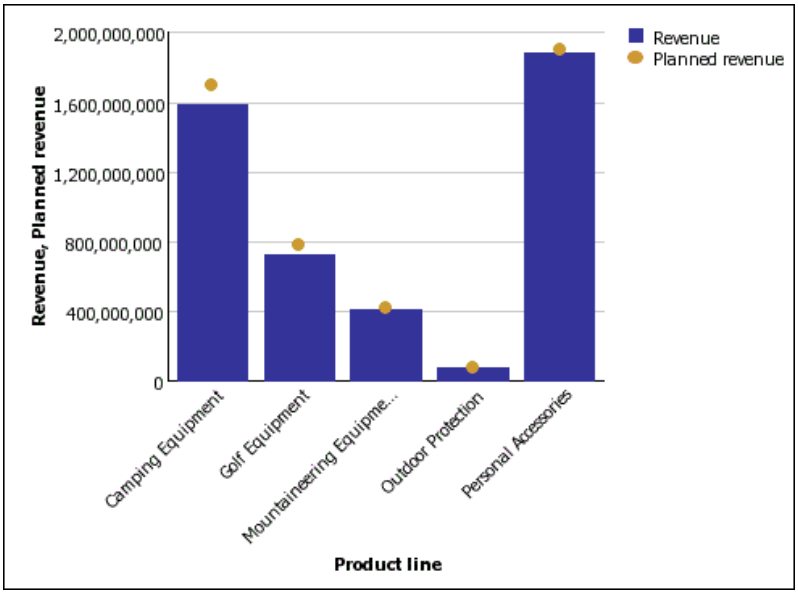
You can convert one chart type into another. We will start with a progressive column chart to show revenue contribution by product line.

Steps to Convert a Chart

1. Create a new chart.
2. In the **Chart group** pane, click **Progressive**.
3. In the **Chart type** pane, click **Progressive Column**.
4. Click **OK**.
5. Drag **Revenue** (in **Sales fact**) to the **Default measure** (**y-axis**) drop zone.
6. Drag **Product line** (in **Product**) to the **Category** (**x-axis**) drop zone.
7. Run the report to see your chart.
8. Return to Report Studio.
9. Right-click the chart background and click **Convert Chart**.
10. Click **OK** to open the default column chart.

Up to now, we have used only a single measure on the chart. Now we will use two measures to compare revenue versus planned revenue for each product line.

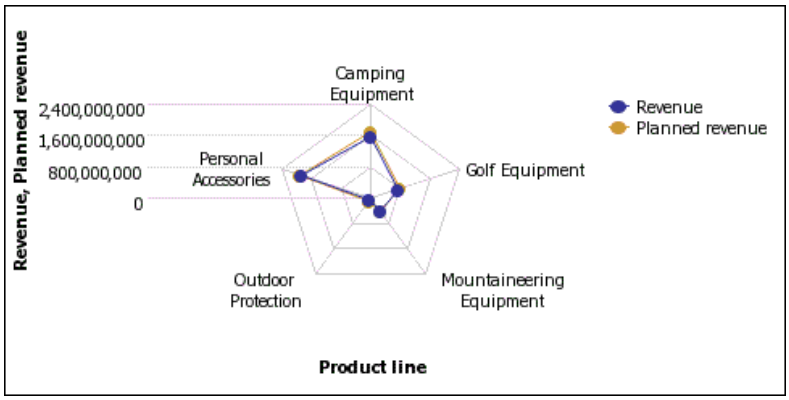
11. Drag **Revenue** from the **Default measure (y-axis)** drop zone to the **Series** drop zone.
12. Drag **Planned Revenue** (in **Sales fact**) to the **Series** drop zone, under **Revenue**.
13. Click the **Planned Revenue** legend item icon, and, in the **Properties** pane, set the **Chart Type** property to **Line**.
14. Change the **Line** property from **Yes** to **No**.
15. Run the report to see your chart.



16. Return to Report Studio.

    You can also show **Planned revenue** as a radar chart.

17. Right-click the chart background and click **Convert Chart**.
18. In the **Convert Chart** dialog box, click the **Radar, Polar** chart group and the **Radar with Markers** chart type. Click **OK**.
19. In the **Chart Conversion Loss** dialog box, click **OK**.
20. Run the report to see your chart.



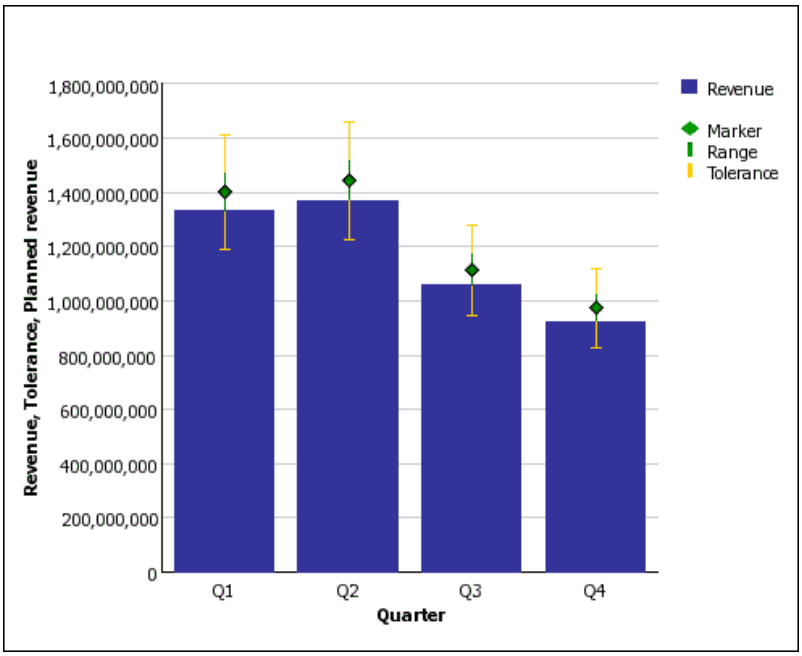
It is difficult to distinguish **Revenue** from **Planned revenue** in this chart type. Another chart you can use to compare revenue and planned revenue is a metrics range chart.

Steps to Create a Metrics Range Chart

1. Create a new chart.
2. In the **Chart group** pane, click **Metrics Range**.
3. In the **Chart type** pane, click **Column Chart with Range Indicators**.
4. Drag the following items to the chart:
   - **Revenue** (in **Sales fact**) to the **Actual (y-axis)** drop zone.
   - **Planned revenue** (in **Sales fact**) to the **Target (marker)** drop zone.
   - **Quarter** (in **Time dimension**) to the **Category (x-axis)** drop zone
5. From the **Insertable Objects** pane, on the **Toolbox** tab, drag a query calculation to the **Tolerance (marker)** drop zone.
6. In the **Create Calculation** dialog box, type

   Tolerance
7. Click **OK**.
8. In the **Expression Definition** box, type the expression:

   [Planned revenue] * 0.1
9. Click **OK**.
10. Click run on the toolbar, and view the report.



Need More Help?

- Add Data to a Report
- Charts
- Working with Queries
- Run a Report

Try It Yourself - Create Map Reports

Use maps to represent data that can be displayed spatially. Cognos 8 includes maps that you can link to data in your data source. You can edit maps and add additional maps by using Map Manager. For more information see the Map Manager *Installation and User Guide*.

Show the Distribution of Revenue by Country

In this exercise, you use a map of the world to show revenue by country.

Steps to Begin a Map

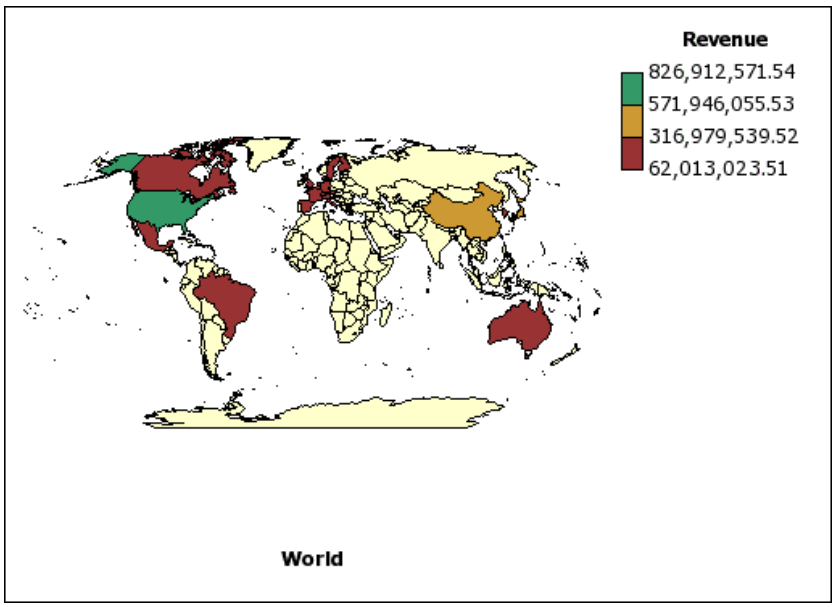
1. Open Report Studio with the **Go Data Warehouse (query)** package.
2. Create a new **Map** report.
3. Select the map of the world.
4. In the **Region Layers** box, click **Countries + Territories**.
5. In the **Point Layers** box, click **None**.

6. Click **OK**.

   Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map object.

Steps to Define the Region Layer

1. Drag the following data items to the map:
   - **Revenue** (in **Sales fact**) to the **Color** drop zone.
   - **Retailer country** (in **Retailer site**) to the **Location** drop zone.
2. Run the report.



Show Revenue and Margins for Cities in the United States

In this exercise you use a map of the United States to show revenue and margins for outlets by using points for cities.

Steps to Begin the Map

1. Open Report Studio with the **GO Data Warehouse (query)** package.
2. Create a new **Map** report.
3. In the **Choose Map** dialog box, in the **Maps** pane, expand **Americas** and click **United States (Continental 48)**.
4. In the **Region Layers** box, click **States**.
5. In the **Point Layers** box, click **Major Cities**.
6. Click **OK**.

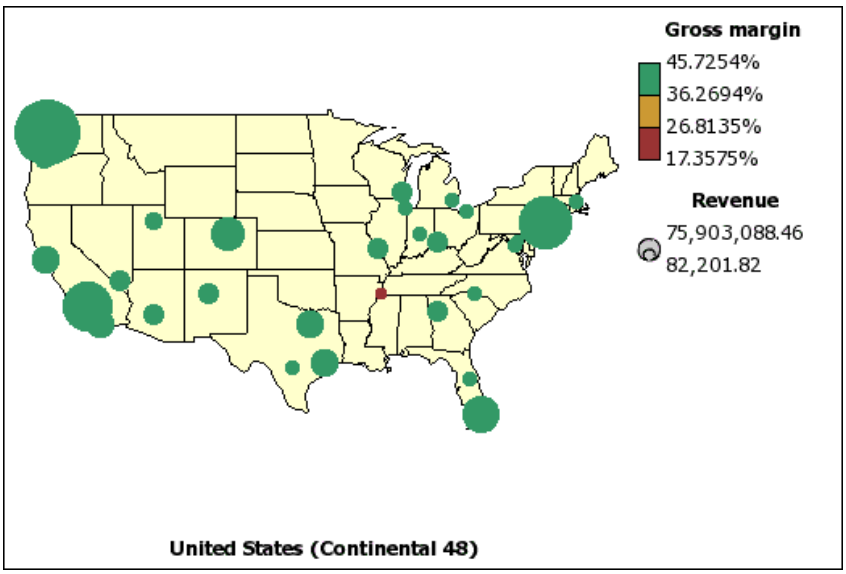
Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map body.

Steps to Set the Point Layer

1. Drag the following data items to the map:
   - **Margin** (in **Sales fact**) to the **Color** drop zone in the **Point layer**.
   - **Revenue** (in **Sales fact**) to the **Size** drop zone in the **Point layer**.
   - **City** (in **Retailer site**) to the **Location** drop zone in the **Point layer**.
2. Click the map object.
3. In the **Properties** pane, click **Ignore Data with no Features** and change the property to **Yes**.

   This specifies that the report can run even if there is not a match in the map file for every data value retrieved from the data source.
4. Run the report.

   The points are located at cities with sales branches. The size of the point shows the level of revenue and the color of the point shows the margin for each sales location.



Steps to Change the Size of the Points

1. In the point layer, click **Revenue** in the **Size** drop zone.
2. In the **General** section of the **Properties** pane, click **Minimum Size**, and select **2pt**.
3. Click **Maximum Size**, and select **10pt**.
4. Run the report.

   The points on the map are now smaller.

Need More Help?

- Parts of a Map Report
- Ignore Data with no Features
- Match Data Values to Names in the Map File
- Add Titles, Subtitles, and Footers
- Add Legend Titles
- Add a Note to a Map
- Customize a Map
- Drill Through to Another Report From a Map
- Add and Edit Maps

Try It Yourself - Create a Multiple-Page Report

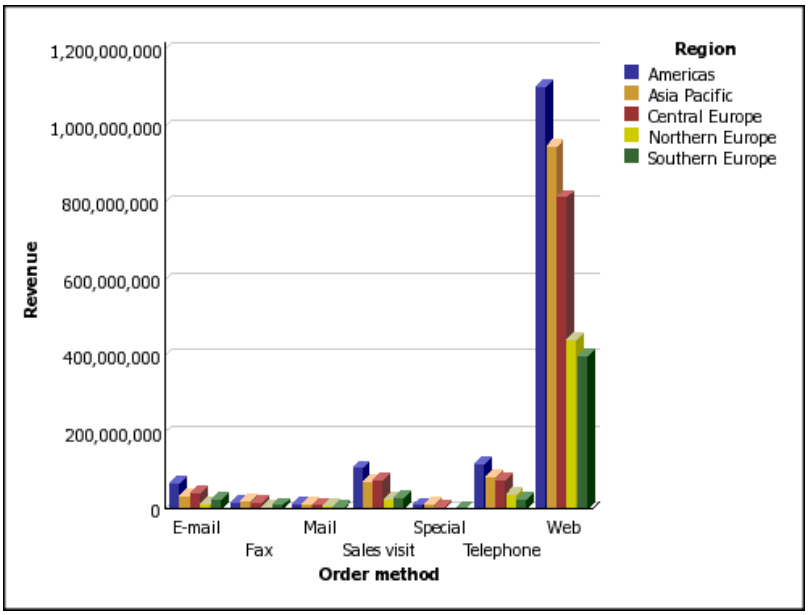
Add pages to a new or existing report that contain different content to create a multiple-page report.

In this topic, you learn how to create a multiple-page report that shows two sales performance reports.

It should take 10 minutes to complete this topic, and the two report pages will look like this.

Performance by Region

Region	Retailer country	Planned revenue	Revenue	Gross profit
Americas	Brazil	144,697,427.54	138,276,679.05	56,889,534.34
	Canada	319,653,059.78	306,159,341.39	125,742,972.08
	Mexico	177,346,683.04	167,187,013.22	68,337,334.67
	United States	866,001,564.74	826,912,571.54	339,535,414.37
Asia Pacific	Australia	115,311,806.09	109,299,969.14	45,045,452.47
	China	334,081,977.75	317,244,910.76	129,827,299.83
	Japan	374,437,885.62	357,446,606.95	147,130,700.53
	Korea	212,939,588.70	200,725,307.09	81,927,957.57
	Singapore	208,460,727.80	197,622,385.04	80,766,644.9
Central Europe	Belgium	118,195,700.30	113,031,617.63	46,659,326.76
	France	300,166,314.17	286,569,502.15	117,788,558.75
	Germany	275,476,018.88	262,313,028.79	107,654,488.42
	Switzerland	104,890,676.14	100,731,875.51	41,876,015.74
	United Kingdom	291,298,414.99	277,509,546.56	113,674,357.53



Steps

1. Create a list report that uses the sample package named **GO Data Warehouse (query)**.
2. Add these data items to the list:
   - **Region** (in **Retailer site**)
   - **Retailer country** (in **Retailer site**)
   - **Planned revenue** (in **Sales fact**)
   - **Revenue** (in **Sales fact**)
   - **Gross profit** (in **Sales fact**)
3. Group the **Region** column to make the report easier to read.
4. Change the blue text placeholder to this text:

   Performance by Region
5. Add a page.
6. Add a 3-D column chart to the new page as a new query.
7. Add these data items to the chart:
   - **Revenue** (in **Sales fact**) to the **Default measure (y-axis)** drop zone
   - **Order method** (in **Order method**) to the **Category (y-axis)** drop zone
   - **Region** (in **Retailer site**) to the **Series** drop zone

   Tip: Use the **Source** tab in the **Insertable Objects** pane.
8. Run the report and click **Page down** to see the two different pages.

Need More Help?

- Add a Page
- Add Data to a Report
- Charts
- Crosstab Reports
- List Reports
- Run a Report

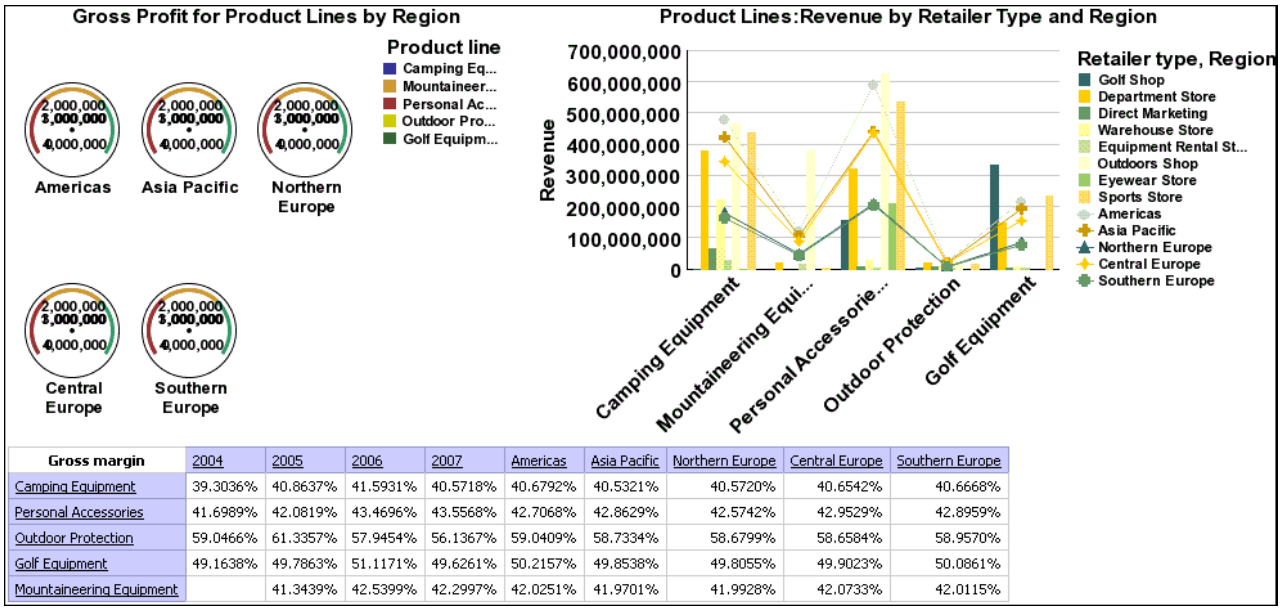
Try It Yourself - Create a Dashboard Report

Create a dashboard report to show a high-level graphical view of company performance indicators. By using dashboards, users can

- drill up and drill down to see higher and lower levels of detail
- navigate to other targets, including Report Studio reports, Query Studio reports, and Analysis Studio analyses
- include multiple charts derived from different data sources in a single report

In this topic, you will learn how to create a dashboard so that users can examine important sales data in a single view.

It should take 15 to 20 minutes to complete this topic, and your dashboard will look like this.



Gross margin	2004	2005	2006	2007	Americas	Asia Pacific	Northern Europe	Central Europe	Southern Europe
Camping Equipment	39.3036%	40.8637%	41.5931%	40.5718%	40.6792%	40.5321%	40.5720%	40.6542%	40.6668%
Personal Accessories	41.6989%	42.0819%	43.4696%	43.5568%	42.7068%	42.8629%	42.5742%	42.9529%	42.8959%
Outdoor Protection	59.0466%	61.3357%	57.9454%	56.1367%	59.0409%	58.7334%	58.6799%	58.6584%	58.9570%
Golf Equipment	49.1638%	49.7863%	51.1171%	49.6261%	50.2157%	49.8538%	49.8055%	49.9023%	50.0861%
Mountaineering Equipment		41.3439%	42.5399%	42.2997%	42.0251%	41.9701%	41.9928%	42.0733%	42.0115%

Steps to Create the Report and Add Data

1. Create a blank report that uses the sample package named **GO Data Warehouse (analysis)**.
2. Add a gauge chart, column chart, and crosstab.

3. Drag the following items to the gauge chart:
   - **Gross profit** (in **Sales fact**) to the **Default measure** drop zone
   - **Product line** level (in **Product**) to the **Needles (series)** drop zone
   - **Region** level (in **Retailer site**) to the **Gauges** drop zone
4. Drag the following items to the column chart:
   - **Revenue** (in **Sales fact**) to the **Default measure (y-axis)** drop zone
   - **Product line** level (in **Product**) to the **Category (x-axis)** drop zone
   - **Retailer type** level (in **Retailer**) and **Region** (in **Retailer site**) to the **Series** drop zone
5. Drag the following items to the crosstab:
   - **Gross margin** (in **Sales fact**) to the **Measures** drop zone
   - **Product line** level (in **Product**) to the **Rows** drop zone
   - **Year** level (in **Time dimension**) and **Region** level (in **Retailer site**) to the **Columns** drop zone

Steps to Format the Gauge Chart

1. Click the gauge chart.
2. Set the **Size & Overflow** property width to **400 px**.
3. Set the **Title** property to **Show**.
4. Double-click the chart title and type the following text:

   Gross Profit for Product Lines by Region
5. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
6. Expand **Axis titles**, click the horizontal axis title, and set the **Visible** property to **No**.
7. Click the numerical axis and, in the **Maximum Value** property, type **4000000**.

Steps to Format the Column Chart

1. Click the **Region** series and set the **Chart Type** property to **Line**.

   Note: The column chart becomes a combination chart because you render the sales territory data as lines.
2. Click the chart, click the **Palette** property, and select **Contemporary** from the **Palette** list.
3. Set the **Size & Overflow** property width to **600 px**.
4. Set the **Title** property to **Show**.
5. Double-click the chart title and type the following text:

   Product Lines: Revenue by Retailer Type and Region

6. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
7. Expand **Axis titles**, click the horizontal axis title, and set the **Visible** property to **No**.

Steps to Specify the Drill Behavior

1. From the **Data** menu, click **Drill Behavior**.
2. Select the **Allow drill-up and drill-down** check box.
3. Run the dashboard to view what it will look like for your users.

   Users can drill up or drill down on values in the report to view related information.

Need More Help?

- Choose a Report Template
- Add Data to a Report
- Working with Dimensional Data
- Specify the Properties of a Gauge Chart
- Specify the Height and Width
- Specify the Font
- Example - Customize the Axis Titles
- Create a Drill-up/Drill-down Report

Try It Yourself - Create a Dynamic Report That Shows Period-to-date Data

When you are working with a dimensional data source, you can use dimensional functions to retrieve data for a specific period-to-date.

In this topic, you learn how to create a dynamic report that retrieves year-to-date revenue for each product line. The report also shows the percentage of the revenue generated in the month users select when they run the report.

It should take 15-20 minutes to complete this topic, and your report will look like this.

Revenue	April	YTD Revenue for April	Percent of YTD Revenue
Camping Equipment	24,764,220	24,764,220	100.00%
Personal Accessories	26,057,935.06	26,057,935.06	100.00%
Outdoor Protection	2,713,436.3	2,713,436.3	100.00%
Golf Equipment	11,742,767.4	11,742,767.4	100.00%
Mountaineering Equipment			

Steps to Create the Report

1. Create a crosstab report that uses the sample package named **Go Data Warehouse (analysis)**.
2. Add the following data items to the report:
   - **Product line** level (in **Product**) to the **Rows** drop zone
   - **Revenue** (in **Sales fact**) to the **Measures** drop zone

   Tip: Use the **Source** tab in the **Insertable Objects** pane.
3. Create this query calculation named **Selected Month** in the **Columns** drop zone:

   [Sales].[Time dimension].[Time dimension].[Month]->?Month?
4. Create this query calculation named **Year to Date Set** next to the **Selected Month** column:

   periodsToDate([Sales].[Time dimension].[Time dimension].[Year],[Selected Month])

   Tip: For information about the **periodsToDate** function, click the function in the **Functions** tab. It is located in the **M-Q** folder in the **Common Functions** folder.
5. Click the **Year to Date Set** column and click the cut button.
6. Create this query calculation named **YTD Revenue** next to the **Selected Month** column:

   total(currentMeasure within set [Year to Date Set])
7. Create this query calculation named **Percent of YTD Revenue** next to the **YTD Revenue** column:

   [Selected Month]/[YTD Revenue]

Steps to Format the Report

1. Click the lock/unlock button to unlock the report.
2. Add this text item to the left of the text in the **YTD Revenue** column, and add a blank space after the text:

   YTD Revenue for
3. Set the font style of the text to italic.
4. In the **YTD Revenue** column, click **<#YTD Revenue#>** and change the **Source Type** property to **Report Expression**.
5. Double-click the **Report Expression** property and drag the **Month** parameter to the **Expression Definition** box.

   Tip: You can find the **Month** parameter in the **Parameters** tab.

6. Click the lock/unlock button to lock the report.
7. Right-click the **Percent of YTD Revenue** column title and click **Fact Cells for "Percent of YTD Revenue"**.
8. In the **Properties** pane, double-click the **Data Format** property.
9. Under **Format type**, click **Percent**.
10. Set the **No of Decimals** property to **2**.
11. Run the report to view what it will look like for your users.

    Users are prompted to select a month. When they click **OK**, the report shows year-to-date revenue for each product line and the percentage of revenue generated for the selected month.

Need More Help?

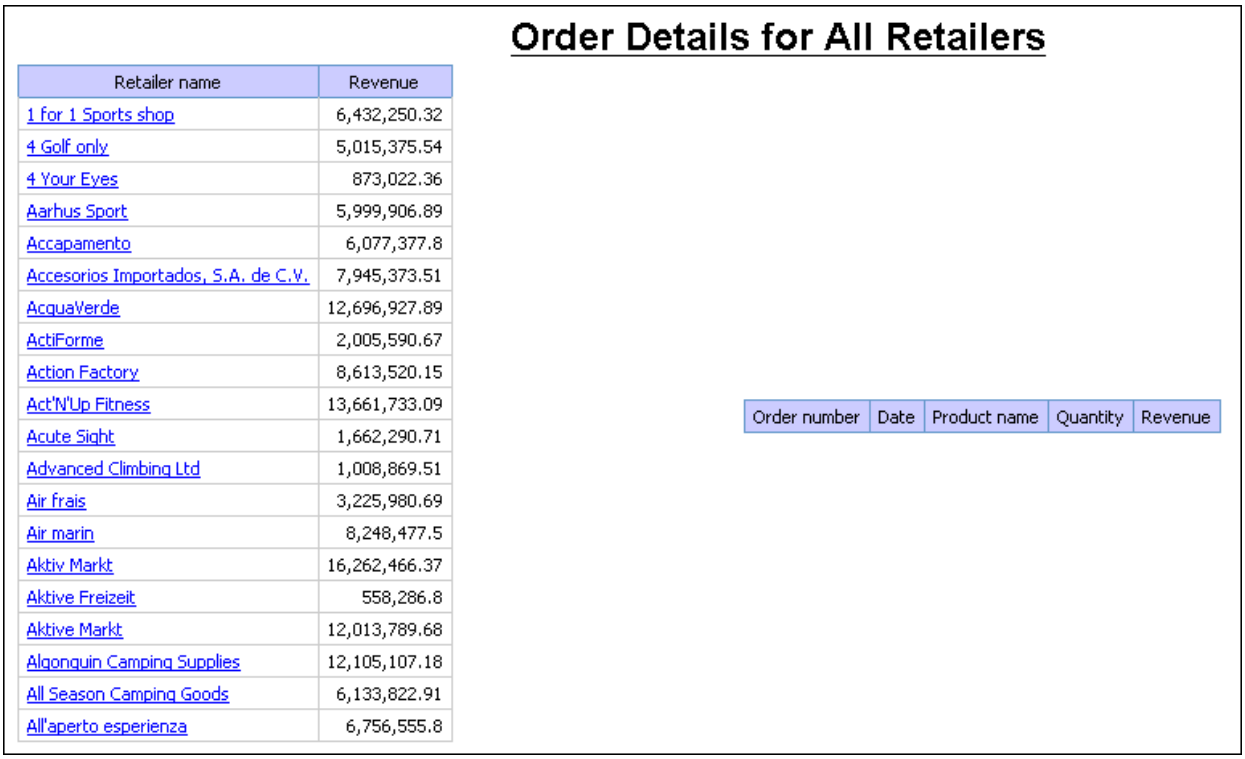
- Choose a Report Template
- Add Data to a Report
- Working with Dimensional Data
- Using the Expression Editor
- Format Crosstabs
- Specify the Font
- Format Data

Try It Yourself - Create a Report with Drill-through Access to Itself

Create a report that drills to itself so that users can view detailed information in the same report.

In this topic, you learn how to create a report that shows revenue by each retailer. Users can access detailed order information for a particular retailer.

It should take 20-25 minutes to complete this topic, and your report will look like this.



Order Details for All Retailers

Retailer name	Revenue
1 for 1 Sports shop	6,432,250.32
4 Golf only	5,015,375.54
4 Your Eyes	873,022.36
Aarhus Sport	5,999,906.89
Accapamento	6,077,377.8
Accesorios Importados, S.A. de C.V.	7,945,373.51
AcquaVerde	12,696,927.89
ActiForme	2,005,590.67
Action Factory	8,613,520.15
Act'N'Up Fitness	13,661,733.09
Acute Sight	1,662,290.71
Advanced Climbing Ltd	1,008,869.51
Air frais	3,225,980.69
Air marin	8,248,477.5
Aktiv Markt	16,262,466.37
Aktive Freizeit	558,286.8
Aktive Markt	12,013,789.68
Algonquin Camping Supplies	12,105,107.18
All Season Camping Goods	6,133,822.91
All'aperto esperienza	6,756,555.8

Order number	Date	Product name	Quantity	Revenue

Steps to Create the Report and Add Data

1. Create a new blank report that uses the sample package named **GO Data Warehouse (query)**.
2. Add a table with two columns and one row.
3. Add a list object to each column in the table.
4. Add these data items to the first list:
   - **Retailer name** (in **Retailer**)
   - **Revenue** (in **Sales fact**)

   Tip: Use the **Source** tab in the **Insertable Objects** pane.
5. Set the sort order for the **Retailer name** column to **Sort Ascending**.
6. Add **Retailer name** in **Retailer** to the second list.
7. Add these data items to the second list:
   - **Order number** in **Sales order**
   - **Date** in **Time Dimension**
   - **Product name** in **Product**
   - **Quantity** in **Sales fact**
   - **Revenue** in **Sales fact**

Tip: To simultaneously add all the data items to the list, press Ctrl+click and select the items before dragging them to the list.

8. In the second list, click the **Retailer name** column and click the cut button.
9. In the second list, create the following parameterized filter:

   [Retailer name]=?Selected retailer?
10. Save the report.

Steps to Set up Drill-through Access

1. In the left list, right-click a **Retailer name** column (and not the column title) and click **Drill-Through Definitions**.
2. Create a new drill-through definition.
3. Specify the report as the target report.
4. Under **Parameters**, click the edit button.
5. Set the **Selected retailer** parameter to pass data item values using the **Retailer name** data item.

Steps to Add a Prompt

1. In the **Toolbox** tab, add a value prompt under the table.

   Because you are setting up drill-through access to the same report, you must add a prompt with a default value so that users are not prompted when they run the report.
2. In the **Prompt Wizard**, specify that the prompt is to use the existing parameter named **Selected retailer**, and click **Finish**.
3. Click the value prompt.
4. In the **Properties** pane, double-click the **Default Selections** property and add a simple selection named **NoRetailer**.
5. Set the following properties:
   - **Required** to **No**
   - **Hide Adornments** to **Yes**
   - **Visible** to **No**

Steps to Add a Report Title

1. Double-click the report title and type the following text, adding a blank space at the end:

   Order Details for
2. In the **Toolbox** tab, add the following layout calculation to the right of the report title:

if(ParamDisplayValue('Selected retailer')='NoRetailer') then 'All Retailers' else (ParamDisplayValue('Selected retailer'))

When the report is run, the report title changes to reflect the retailer selected by the user. If no retailer is selected, 'All Retailers' appears.

3. Click the layout calculation and, in the **Properties** pane, set the class to **Report title text**.

Steps to Add Conditional Formatting

1. Pause the pointer over the condition explorer button and click **Variables**.

   Use conditional formatting to show a message above the second list when users have not selected a retailer from the first list.

2. Create this boolean variable named **HighLevel**:

   ParamDisplayValue('Selected retailer')<>'NoRetailer'

3. Pause the pointer over the page explorer button and click **Page1**.
4. In the **Toolbox** tab, add a block above the second list.
5. Insert a text item in the block with the following text:

   Select a retailer in the left list to view order details below

6. Set the font style for the text to bold.
7. Click the text item and, in the **Properties** pane, set the **Style Variable** property to the **HighLevel** variable.
8. Pause the pointer over the condition explorer and click the **Yes** value for the **HighLevel** variable.
9. Set the **Visible** property for the text item to **No**.
10. Triple-click the explorer bar to turn off conditional formatting.
11. Run the report to view what it will look like for your users.

    A list appears on the left that shows revenue for each retailer. When users click a retailer, order information for the selected retailer appears in the second list.

Need More Help?

- Use Tables to Control Where Objects Appear
- Add Data to a Report
- Create a Parameter to Produce a Prompt
- Set Up Drill-through Access in a Report
- Create a Prompt Directly in a Report Page
- Create a Calculation

- Using the Expression Editor
- Add a Variable
- Hide and Show Objects

Chapter 8: Upgrading Reports

When you open a report that was created in a previous version of Report Studio, it is automatically upgraded. Any problems detected during the upgrade process appear in the **Upgrade Information** dialog box. You must fix any errors in the report and then validate the report before you can run it. If only warnings and information appear in the dialog box, these will disappear when you click **OK**.

Tip: To view this dialog box again, from the **File** menu, click **Upgrade Information**.

Once a report is upgraded to the most recent version of Cognos 8, you can no longer open it with a previous version of Cognos 8.

For more information about changes in the product behavior, see **Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior** on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Limitations When Upgrading Reports from Cognos ReportNet

When you upgrade Cognos 8, some reports may look or behave differently after the upgrade. The upgrade does not account for the following:

- undocumented and unsupported features such as JavaScript that refers to Cognos HTML objects

  If advanced report authors used such features, they may have to recreate the features to complete the upgrade of the report.

- reports that do not include a sort order

  Data may appear in a different order after upgrading. If sort order is important, ensure that the report or model specifies a sort order before upgrading.

- layout errors after upgrading from ReportNet

  Layout errors are suppressed by default in ReportNet. Cognos 8 does not suppress layout errors. Users may have to correct or remove report layout expressions that cause errors.

- changes in report format between ReportNet and Cognos 8

  If you used the default format in ReportNet, upgraded reports will retain that format. However, new reports in Cognos 8 will use the Cognos 8 format. If you want a consistent style across all reports, you must edit the styles property in each report and select or deselect the Use 1.x report styles option.

- customized style sheets in ReportNet or earlier versions of Cognos 8

  If you edited the ReportNet style sheet (default_layout.css) or the Cognos 8 stylesheet (globalreportstyles.css), reports will lose the formatting after upgrading. You must reapply the

changes to the Cognos 8 stylesheets and copy the stylesheets to the Cognos 8 server and the Web server.

- PDF reports in ReportNet

  PDF rendering in Cognos 8 behaves like HTML rendering. After upgrading, PDF reports may have different font size, column wrapping, or word wrapping. You may have to change each affected report or change the default font in the Cognos 8 style sheet. For more information, see "Formatting a Report" (p. 133).

- reports that use IF-THEN-ELSE statements in ReportNet

  If you use assignments of different data types after THEN and ELSE in ReportNet, the reports will generate invalid coercion errors after upgrading. You may have to recast the variables or change the assignments in the affected reports.

- solve order

  In previous versions of Cognos 8, you could specify the solve order for objects such as crosstab node members (p. 86). Solve order is now specified for data items. When you upgrade a report, solve orders specified in the report are moved to data items. For more information about solve order, see "Multiple Calculations in Crosstabs and Charts" (p. 191).

- changes in chart behavior from ReportNet to Cognos 8

  In ReportNet, if a chart is created with a user-specified minimum value and all data values are below the minimum value, the chart starts at the user-specified minimum value and contains no data.

  In Cognos 8, the same parameters result in a chart that ignores the user-specified minimum value and uses a range that includes all of the data values for the chart.

- database only processing of queries

  If you specified in ReportNet that the processing for a query should be **Database Only**, when you upgrade your report to Cognos 8, some processing may now occur locally.

Changes in the Behavior of Count Between Cognos ReportNet(R) 1.1 and Cognos 8

In Cognos ReportNet 1.1, double counting could occur when applying Count or Count Distinct to a query item. This occurred primarily when querying multiple query subjects that were joined 1-to-N in the model if the counted item existed in the query subject on the 1 side of the join. The explanation is that Count or Count Distinct was applied after the join operation.

Count Distinct worked only when the item to which Count Distinct was applied was the column used in the join. In this case, the column to be counted could have identical values for the different values used in the join condition.

In Cognos 8, improvements to the count functionality cannot be handled through an automatic upgrade. The new approach avoids double counting on the 1 side of a 1-to-N join. The explanation is that Count or Count Distinct is now applied before the join operation.

There is now a lesser need to use Count Distinct and, when used, it will be successful more often. Count Distinct is no longer required to overcome double counting. Instead, it can be used as intended to select distinct values that exist in a query subject.

When you want to count repeated occurrences of a value, we recommend that you do one of the following:

- Apply a count operation on a column based on a query in which an explicit join occurs.

  This applies the count after the join.

- Count rows in a report by using a layout calculation object (p. 143) or by counting the literal value 1.

Upgrading Reports from Cognos 8 Version 8.1 or 8.2

When you upgrade Cognos 8 to version 8.3, some reports may look or behave differently after the upgrade. This section describes changes that you may encounter in your reports.

Product Behavior After Upgrade

When you upgrade from Cognos 8 BI version 8.2 to version 8.3, some features in Cognos 8 may behave differently after the upgrade. When you upgrade reports, for example, changes in behavior may cause validation errors. Documentation is available about the behavior changes. This documentation includes examples of the changed behavior and solutions for issues that may occur during the upgrade. For more information, see *Upgrading to Cognos 8 BI 8.3: Changes in Product Behavior* on the Global Customer Services Web site (http://support.cognos.com/go/docs/cognos8_bi/8.3/changes_productbehavior.html).

Members Containing No Data Are Not Suppressed for SAP BW

If you upgrade a crosstab report that uses a SAP BW data source and includes calculations, filters, or nesting, the suppression of null values may not occur as expected. You may see additional empty rows and columns.

By default, null values are suppressed for list and grouped list reports.

To effectively remove null values in crosstabs that use a SAP BW data source, insert individual members to create the report. You can also ensure that SAP BW members are assigned to proper dimension hierarchies and levels within the BW cube.

Thousands Separators Missing

You may encounter missing thousands separators in your reports in the following circumstances:

- The report contains an item that does not specify an explicit data format.
- The report item refers to another item in the same report.

- That second report item refers to an item in the Framework Manager model with **Usage** property set to **Identifier** or **Attribute**.
- The model item does not specify an explicit format.

To restore the data formats, specify an explicit format either in the report item properties or in the model item properties.

Appendix A: Troubleshooting

This chapter describes some common problems you may encounter. For more troubleshooting problems, see the Troubleshooting section of the *Administration & Security Guide*.

Problems Creating Reports

The topics in this section document problems you may encounter when creating reports.

Division by Zero Operation Appears Differently in Lists and Crosstabs

If you have a list that accesses a relational data source, a calculation containing a division by zero operation appears as a null value, such as an empty cell. In a crosstab, the division by zero operation appears as /0. This happens when the **Avoid Division by Zero** property is set to **Yes**, which is the default.

To have a consistent display of null values in lists and crosstabs, define an if-then-else statement in the expression in the crosstab cell that changes the value /0 to the value null.

Application Error Appears When Upgrading a Report

When upgrading a report, the following error appears if the report contains data items in the page layout that are not in a data container:

RSV-SRV-0040 An application error has occurred. Please contact your Administrator.

This error occurs when Cognos 8 cannot determine the query reference for a data item. Such data items are identified by a small red circle with a white x icon that appears in the lower left corner.

To correct the error, drag the data items into a container. If the container is a list, we recommend that you drag the data items into the list page header or footer, or the overall header or footer. If you want to see the first row of the item on each page or in the overall report, drag the item to the list page header or overall header. If you want to see the item's last row on each page or in the overall report, drag the item to the list page footer or overall footer. **Tip**: If a header or footer does not exist, create it.

Background Color in Template Does not Appear

When creating a Query Studio template in Report Studio, if you add a list object and change its background color, the color change does not appear when you apply the template to a Query Studio report.

To work around this issue, do one of the following:

- Edit the style sheet (CSS) classes for lists in Report Studio.

- Do not add any objects to the page body when you are creating a Query Studio template.
- Leave the page body blank.

Subtotals in Grouped Lists

When using a PowerCube that contains a ragged hierarchy, if you group on the first level in the hierarchy, subtotals may appear in the wrong place or show wrong values.

To resolve the issue, group on the second level.

Chart Labels Overwrite One Another

In Report Studio and Query Studio, if you define a chart and render it in HTML or PDF format using the default sizes, the axis labels of the chart may overwrite each other.

To avoid this problem, make the chart wider or taller by modifying the height and width properties of the chart or enable the Allow Skip property.

Chart Shows Only Every Second Label

You create a report that includes a chart. The **Allow Skip** option is set to false, but when you run the report, labels are skipped.

This can occur if there is not enough room for all labels and the options **Allow 45 Degree Rotation**, **Allow 90 Degree Rotation**, and **Allow Stagger** are also set to false. Cognos 8 has no options for making the labels fit, so it skips every second label.

The solution is to select either **Allow 45 Degree Rotation**, **Allow 90 Degree Rotation**, or **Allow Stagger**.

Chart Gradient Backgrounds Appear Gray

In Report Studio, you can define a custom palette for a chart that includes a gradient. When the chart is rendered in HTML format, the chart background appears gray. This is a Microsoft Web browser issue.

To avoid this problem, select the chart and define the color white as the chart background.

For more information, see the Microsoft Knowledge Base article at http://support.microsoft.com/kb/294714.

Problems Filtering Data

You can use filters to specify the subset of records that the report retrieves. Any data that does not meet the filter criteria is eliminated from the report.

The topics in this section document problems you may encounter when using filters in reports.

In general, when creating filters, select functions that are available from the Vendor Specific list. You must use Tabular SQL for other functions.

Tip: If combining internal functions and vendor functions, set processing to **Limited Local**.

HRESULT= DB_E_CANTCONVERTVALUE Error When Filtering on a _make_timestamp Column

You cannot filter on a _make_timestamp column, and the following error messages appear:

UDA-SQL-0114 The cursor supplied to the operation "sqlOpenResult" is inactive

UDA-SQL-0206 The OLEDB driver returned the following value: HRESULT= DB_E_CANTCONVERTVALUE "

RSV-SRV-0025 Unable to execute this request

The solution is to apply the filter after aggregation and not before.

Problems Running Reports

The topics in this section document problems you may encounter when viewing or running reports.

Summaries in Report Do not Correspond to the Visible Members

If a crosstab or chart created in Report Studio using a dimensional data source has a context-dependent set function such as `filter` or `topCount` on an edge, summaries do not correspond to the visible members. This occurs when the summaries use the `within set` aggregation mode.

This problem occurs because a summary that uses the `within set` aggregation mode uses a set that is dependent on the members that it intersects with on the opposite edge. For example, the following crosstab has as columns the top three products returned. The expression used to generate the columns is

`topCount`([Product],3,[Return quantity])

where [Product] is the level.

Return quantity	BugShield Lotion 89110	BugShield Extreme 90110	Sun Shelter 30 94110	Total(ReturnedProducts)	Minimum(ReturnedProducts)
Americas	25,219	19,870	13,814	**62,392**	**17,303**
Asia Pacific	22,822	19,171	6,389	**54,758**	**12,765**
Northern Europe	8,325	14,634	4,065	**32,936**	**8,325**
Central Europe	17,627	13,854	14,089	**45,570**	**13,854**
Southern Europe	7,196	4,726	5,401	**20,220**	**5,790**
Total(Region)	**81,189**	**72,255**	**43,758**	215,876	58,037
Minimum(Region)	**7,196**	**4,726**	**4,065**	20,220	5,790

The summary values for **Total**(**ReturnedProducts**) and **Minimum**(**ReturnedProducts**) for all rows except **Central Europe** do not correspond to the member values in the crosstab. This means that the top three products returned in all regions except for Central Europe are not Bug Shield Lotion 89110, Bug Shield Extreme 90110, and Sun Shelter 30 94110. Note that the summary values for **Total**(**Region**) and **Minimum**(**Region**) do correspond to the visible member values. That is because those summary values represent the total and minimum quantities returned for those three products in each region.

You can see what the top three products returned in each region are by dragging the columns to the right of the rows, creating the following single-edge crosstab.

Americas	*BugShield Lotion 89110*	*25,219*
	BugShield Extreme 90110	*19,870*
	TrailChef Water Bag 1110	*17,303*
	Total(ReturnedProducts)	**62,392**
	Minimum(ReturnedProducts)	**17,303**
Asia Pacific	*BugShield Lotion 89110*	*22,822*
	BugShield Extreme 90110	*19,171*
	Single Edge 73110	*12,765*
	Total(ReturnedProducts)	**54,758**
	Minimum(ReturnedProducts)	**12,765**
Northern Europe	*BugShield Extreme 90110*	*14,634*
	Star Peg 16110	*9,977*
	BugShield Lotion 89110	*8,325*
	Total(ReturnedProducts)	**32,936**
	Minimum(ReturnedProducts)	**8,325**
Central Europe	*BugShield Lotion 89110*	*17,627*
	Sun Shelter 30 94110	*14,089*
	BugShield Extreme 90110	*13,854*
	Total(ReturnedProducts)	**45,570**
	Minimum(ReturnedProducts)	**13,854**
Southern Europe	*Single Edge 73110*	*7,234*
	BugShield Lotion 89110	*7,196*
	TrailChef Cup 4110	*5,790*
	Total(ReturnedProducts)	**20,220**
	Minimum(ReturnedProducts)	**5,790**

To obtain summary values that reflect the visible members, modify the expression of the data item containing the context-dependent set function so that it includes a tuple that is locked to the default member of every hierarchy that appears on the opposite edge. For this example, modify the expression to the following:

`topCount`([Product],3,`tuple`([Return quantity], `defaultMember`([Retailer site])))

where [Product] is the level and [Retailer site] is the hierarchy.

When you run the report, all summary values reflect the visible members in the crosstab.

Return quantity	*BugShield Lotion 89110*	*BugShield Extreme 90110*	*Sun Shelter 30 94110*	Total(ReturnedProducts)	Minimum(ReturnedProducts)
Americas	*25,219*	*19,870*	*13,814*	**58,903**	**13,814**
Asia Pacific	*22,822*	*19,171*	*6,389*	**48,382**	**6,389**
Northern Europe	*8,325*	*14,634*	*4,065*	**27,024**	**4,065**
Central Europe	*17,627*	*13,854*	*14,089*	**45,570**	**13,854**
Southern Europe	*7,196*	*4,726*	*5,401*	**17,323**	**4,726**
Total(Region)	***81,189***	***72,255***	***43,758***	**197,202**	**42,848**
Minimum(Region)	***7,196***	***4,726***	***4,065***	**17,323**	**4,065**

Cannot Find the Database in the Content Store (Error QE-DEF-0288)

You cannot retrieve data from the selected database when running a report from Query Studio, Cognos Connection, or Report Studio.

The following error message appears:

QE-DEF-0288 Unable to find the database...

If this error does not occur when you are logged on as an administrator, then to solve the problem, ensure that the user has permissions to the signon embedded. If this error always occurs, the data source has not been created. Create the data source with the name mentioned in the error message.

Parse Errors When Opening or Running an Upgraded Report

Earlier versions of ReportNet and Cognos 8 included the cast_Date function for reports that run on an Oracle database. This function does not exist for Oracle in Cognos 8.1.2 MR1 and later versions. If a report that uses an Oracle database includes the cast_Date function, parse errors will be received when you try to open or run the report.

Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters

In a crosstab report, values support a maximum of 19 characters, including the decimal point. If a value exceeds 19 digits, an overflow error occurs. By default, the decimal precision is set to 7 digits, which restricts the number of integers to 11 digits.

To use more than 11 integers, you must edit a configuration file to reduce the decimal precision. For more information, see the Cognos 8 *Administration and Security Guide*.

The ORA-00907 Error Appears When Running a Report

When using an Oracle 9.2 data source, under certain circumstances, multiple or nested join operations may fail and produce the following error:

ORA-00907: missing right parenthesis

A query that uses both a left outer join and an ON clause condition returns zero values instead of null values.

A Report or Analysis Does Not Run Because of Missing Items

You attempt to run a report or analysis and a message indicates that one or more items are missing or changed. Each missing item is listed by its MUN (member unique name). The MUN includes the compete path within the hierarchy for the item. When you place your cursor on an item in the **Insertable Objects** pane, the MUN for that item is displayed in a tooltip. This situation may occur if members have been removed from or changed in the data source. It may also occur when you attempt to run a report that uses items to which you do not have access. For example, an administrator may create an analysis that includes items that you don't have the correct permission to access.

The solution is to find a suitable replacement in the **Insertable Objects** pane, and drag it to the **New Item** column. The report or analysis will then run.

Cannot View Burst Report

When you burst a report, each burst output is sent to the associated list of recipients. If a list of recipients contains invalid entries, the following occurs:

- The burst output is not saved to Content Manager.

Consequently, you cannot view the burst output in Cognos Connection.

- If you choose to send the output by email, only valid recipients will receive an email. Although the output is sent as an attachment if you select the **Attach the report** check box, no link is generated if you select the **Include a link to the report** check box.
- The following error message appears in the run history for the report, where parameter 1 is the burst key, parameter 2 is the list of recipients, and parameter 3 contains the error messages returned by Content Manager:

  An error occurred while saving the output for the burst instance <param type="string" index="1"/> with the recipients (<param type="string" index="2"/>). Here are the details: <param type="string" index="3"/>

  Note: The list of recipients includes both the valid and invalid recipients.

For example, a report is set up to burst on Country, and the recipients are managers. Running the report produces the following countries and recipients:

- Canada: John, Mary
- US: Peter, Frank
- France: Danielle, Maryse

Frank is an invalid recipient. The burst outputs for Canada and France are saved to Content Manager, but not the U.S. output. If you choose to send an email to each recipient and you selected the **Include a link to the report** check box, the email to Peter will not contain a link to the output for US. The error message that is generated will contain Peter and Frank as values for parameter 2 with no indication as to which is invalid.

To resolve the problem, do the following:

Steps to Correct or Remove Burst Recipients

1. View the error message in the run history for the report.
2. From the list of recipients, determine which recipients are invalid.

   You may need to consult with your administrator to find out which recipients are invalid.
3. Correct or remove the invalid recipients.

   Correcting or removing invalid recipients will depend on how the list of recipients was defined, such as through a calculated field or a burst table.
4. Run the report again.

A Report Upgraded from ReportNet Does Not Retain its Original Look

When you upgrade a report to Cognos 8, a new style sheet is applied that changes the look of the report.

To preserve the formatting that was used in the original report, you can select a different style sheet. This retains the original look of the report and specifies that any new items added to the report, such as list columns or crosstab levels, have the original formatting applied to them.

Steps

1. In Report Studio, from the **File** menu, click **Report Properties**.
2. Select the **Use 1.x report styles** check box.

Drill-through Links Not Active in the Safari Browser

When viewing a PDF report in the Macintosh Safari browser, you cannot open hyperlinks. This is because the macintosh Safari browser does not have the necessary Adobe Acrobat plug-in.

To avoid this problem, use the HTML format when creating drill-through reports that may be viewed in Safari.

A Running Total in Grouped Reports Gives Unexpected Results

You have a running total calculation in a grouped report that returns unexpected values.

Because tabulation of the running total calculation depends on the order in which the grouping is executed, you must ensure that the grouped totals are tabulated before applying the running total.

To ensure that the grouping is executed in correct order, define a running total calculation as a freestanding calculation outside the query subject in Framework Manager, and ensure that the Regular Aggregate property is set to Automatic.

This may also be an issue with other running, moving, and ranking aggregations.

Problems When Drilling Through

The topics in this section document problems you may encounter when drilling through reports.

Cannot Drill Between PowerCubes Because MUNs Do Not Match

A Cognos PowerCube has category codes and source values that are different. Its source values are conformed with another PowerCube however the category codes in both PowerCubes are different. You cannot drill between shared dimensions in these PowerCubes because their Member Unique Names (MUN) do not match. The source value for a PowerCube member is used as the member business key in Cognos 8. If the source value business key is conformed between PowerCubes, you can pass the business key in a calculation to drill through between two cubes where the MUN does not match.

This technique works only if the business keys are unique. If the business keys are not unique, then the result may be unexpected. For example, drilling through from London, England in the source may produce a drill-through report that contains both London, England and London, Ontario.

If drilling through was not enabled at build time, the source cube will pass the category code instead of the source value.

To pass the business key to the target, do the following:

- Create the target report parameter by typing:

  roleValue('_businessKey',[member])=?prompt?

When you drill through, the business key is passed.

For more information about drill-through access, see the Cognos 8 *Administration and Security Guide*.

Unexpected Results When Drilling from a Cube to a Relational Data Source

When you drill from a cube to a relational data source, sometimes no value is found. Other times, the wrong data item appears. The reason for this is that the business key values in the two data sources do not match. Note that the source value for a PowerCube member is used as the business key in Cognos 8 studios.

If the target data source has a query item, for example display name, that corresponds to a member property in the source cube, for example caption, you can pass the caption from the cube to the relational target. You must also modify the drill-through definition object by replacing the original member mapping with the new one. This technique is only supported when the drill through definition is authored in Report Studio.

The business key is passed from the cube source only if the target is a value parameter, and the cube source data item is one of the following:

- business key
- member or member set

Steps to Pass the Caption to the Target

1. In the cube source report, create a calculation that retrieves the caption of the member. Type the following

   caption([***member***])

2. Ensure that the target report has its parameter on the data item, and not on the business key.

When you drill through, instead of the business key, the selected value is passed to the target.

For more information about drill-through access, see the Cognos 8 *Administration and Security Guide*.

Cannot Drill From a Relational Source to a Cube

You cannot drill through from a relational data source to a cube. This is because a cube expects a Member Unique Name (MUN) as a parameter value and relational sources do not use MUNs.

Members have several properties such as a business key and a caption. If either of these match data items within the relational source, drilling through can be performed as long as the cube target report is authored in Report Studio.

If the source data source has a query item, for example display name, that corresponds to a member property in the target cube, for example caption, you must create the parameter on the caption in the target report.

To pass the data item to the cube target, do the following:

- In the cube target report, create a parameter that accepts the caption of the member. Type the following

  caption([***member***])**=?*prompt*?**

When you drill through, the selected value is passed to the target.

For more information about drill-through access, see the Cognos 8 *Administration and Security Guide*.

Appendix B: Samples

Sample reports are included with Cognos 8. When installed, you can find them in the Public Folders tab in Cognos Connection.

Samples in the GO Data Warehouse (analysis) Package

The following reports are some of the reports found in the GO Data Warehouse (analysis) package.

Customer Returns and Satisfaction

This report shows the customer satisfaction survey results for Asia Pacific in 2007. It highlights the customers who are the least satisfied. It also provides information about customers with the highest number of product returns. This report uses the following features:

- bar and column charts (p. 94)
- customizing the color and size of a chart (p. 96)
- list reports (p. 79)
- formatting a list (p. 84)
- conditional highlighting (p. 254)
- templates (p. 75)
- filters (p. 172)

Employee Training by Year

This report shows employee training data for the selected year and quarter(s). A bar chart shows training costs by region and a crosstab shows data for the selected quarter(s). This report uses the following features:

- slicers (p. 174)
- cascading prompts (p. 199)
- bar charts (p. 94)
- customizing the color of a chart (p. 96)
- crosstabs (p. 85)
- report expressions in the subtitle (p. 189)

Great Outdoors Company Balance Sheet as at Dec 31, 2006

This report shows a simple balance sheet with assets, liabilities, and equity for 2006 with a 2005 comparative. It uses the Report Studio Express authoring mode. The Getting Started guide provides a step-by-step example of how to create this report.

Global Bonus Report

This list report shows employees who received more than $2,500 bonus in a year by region. It is grouped by country. It also shows how much the sales target was exceeded for each region. This report uses the following features:

- list reports (p. 79)
- page sets (page breaks by country with different sorting and grouping) (p. 152)
- multiple prompts and parameters (p. 207)
- report expressions in the subtitle (p. 189)
- filters (p. 172)
- conditional highlighting (p. 254)
- hidden objects (p. 261)

Succession Report

This report shows the succession data by department and status for percent ready in a column chart. It also contains a detailed crosstab for the managers associated with the possible successors. This report uses the following features:

- multiple pages (p. 151)
- drilling through to the Manager Profile report (p. 267)
- multiple queries (p. 217)
- filters (p. 172)
- slicers (p. 174)
- lists (p. 79)

Manager Profile

This report shows information about managers, including salary, bonuses, and all compensations grouped by year. This report uses the following features:

- charts (p. 94)
- list reports (p. 79)
- prompts (p. 199)

Planned Headcount

This chart report shows headcount variance compared to the plan for each organization. It is prompted by year. This report uses the following features:

- progressive column charts (p. 340)
- prompts with a default selection directly in a report page (p. 206)
- templates (p. 75)
- hidden objects (p. 261)

Positions to Fill

This report shows a list of department names, positions, longest days to fill the positions, and ranking. The report uses a prompt for the year and is a drill-through target for the Recruitment report.

Recruitment Report

This report show which recruitment method is most efficient for recruiting for certain positions or organizations. This report uses the following features:

- drilling through to the Positions to Fill report (p. 267)
- crosstabs (p. 85)
- prompts (p. 199)
- charts with a customized color palette (p. 96)
- floating object adjustment (p. 162)

Returned Items

This report shows the number of returned items by return reason and retailer type. A column chart shows returned items by product line and region for the selected date range. This report uses the following features:

- date and time prompts (p. 201)
- crosstabs (p. 85)
- bar charts (p. 94)

Returns by Failed Orders in 2006

This report shows quality measures based on product returns and focuses on failed orders. This report uses the following features:

- pie charts (p. 94)

- crosstabs (p. 85) with text indentation
- singletons (p. 196)
- drilling through to the Returns by Damage, Failed Orders and Complaints in 2006 report (p. 267)
- calculations, including the tuple() function (p. 189)

Returns by Damage, Failed Orders and Complaints in 2006

This report shows quality measures based on product returns. This report uses the following features:

- pie charts (p. 94)
- crosstabs (p. 85) with text indentation
- singletons (p. 196)
- calculations, including the tuple() function (p. 189)
- drilling through to the Returns by Failed Orders in 2006 report (p. 267)

Revenue by GO Subsidiary 2005

This prompted chart report shows 2005 quarterly revenues for each GO subsidiary. This report uses the following features:

- templates (p. 75)
- custom palettes
- prompts (p. 199)
- hyperlinks
- customizing charts (p. 96)
- singletons (p. 196)

Returns by Order Method - Prompted Chart

This report shows product returns and reasons filtered on order methods. The Getting Started guide provides a step-by-step example of how to create this report. This report uses the following features:

- bar charts (p. 94)
- prompts (p. 199)
- crosstabs (p. 85)
- filters (p. 172)

Samples in the GO Data Warehouse (query) Package

The following reports are some of the reports found in the GO Data Warehouse (query) package.

Bursted Sales Performance Report

This list report shows how to burst a product sales report to a sales manager for Northern Europe sales staff. To successfully burst this report, Cognos 8 must be configured to use an email server. This report uses the following features:

- list reports (p. 79)
- bursting (p. 246)
- conditional highlighting (p. 254)
- templates (p. 75)
- filtering (p. 172)
- calculations (p. 189)

Samples in the GO Sales (query) Package

The following reports are some of the reports found in the GO Sales (query) package.

Order Invoices - Donald Chow, Sales Person

This report generates invoices for all the sales by Donald Chow. This report uses the following features:

- list reports (p. 79)
- adding list row cells (p. 135)
- calculations (p. 189)
- formatting tables (p. 147)

Great Outdoors Samples

The Great Outdoors samples illustrate product features and technical and business best practices. You can also use them for experimenting with and sharing report design techniques, and for troubleshooting.

For examples related to different kinds of businesses, see the product blueprints on the Cognos Web site (http://www.cognos.com). For information about specific installation choices and environments, see the *Architecture and Deployment Guide*, or the Proven Practices and the Cognos

Implementation Roadmaps on the Cognos Global Customer Services Web site (http://support.cognos.com). For information about audit samples, see the *Administration and Security Guide*.

Where to Find the Samples

The samples are included with the product and the samples for each studio are described in the related user guide and online help. To use the samples, you must set up and configure them, or contact your administrator to find out where they are installed. For instructions on how to set up and configure samples, see the *Installation and Configuration Guide* or the *Administration and Security Guide*.

What Samples Exist

The samples consist of

- two databases that contain all corporate data, the related sample models for query and analysis, and sample cubes, reports, queries, query templates, and dashboards
- a metrics database and the associated metrics, including a strategy map for the consolidated company

  Note: You must have Metric Studio installed to use these samples.
- reports that contain extended tips to explain how and why they were created

  Note: These reports are stored in a folder labeled Practical Examples in the samples folder of the Cognos 8 installation.

Security

You can add security to the samples using an employee list included with the product. The list is in the form of an LDIF file that can be imported into any directory server, such as the Sun Java System Directory Server.

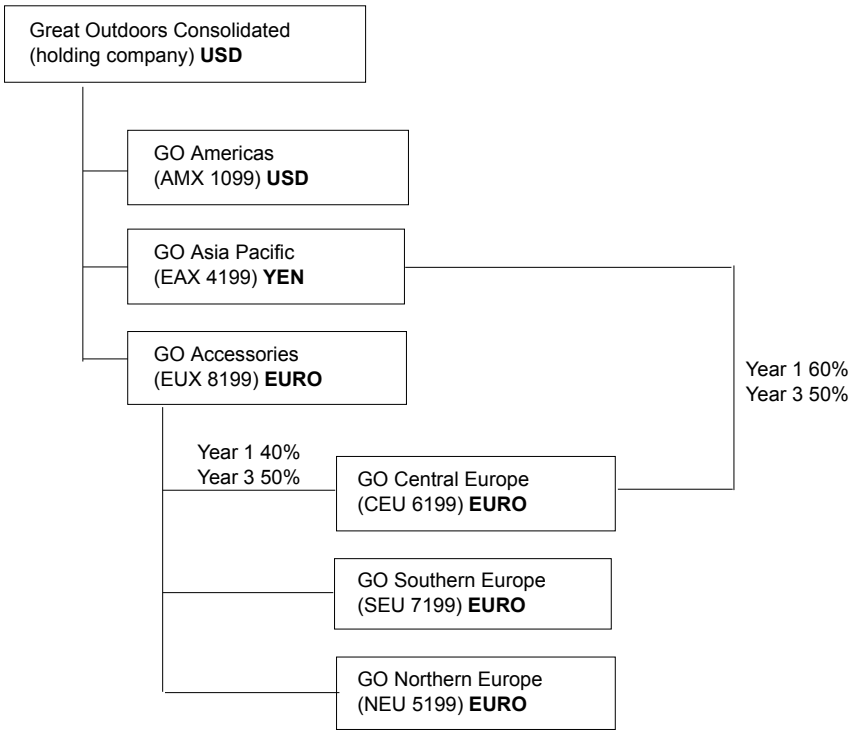
Note: You can download a version of the Sun Java System Directory Server from the Sun Web site (http://www.sun.com/download/). For more information, see the *Installation and Configuration Guide*. For information about conformance, see the Cognos Global Customer Services Web site (http://support.cognos.com).

The Great Outdoors Group of Companies

To make designing examples faster, especially financial examples, some general information about the fictional Great Outdoors company is useful. To look for samples that use particular product features, see the individual sample descriptions in this appendix.

Revenue for The Great Outdoors comes from corporate stores and from franchise operations. The revenues are consolidated from the wholly-owned subsidiaries. The corporate structure is shown below.

The diagram below illustrates the consolidated corporate structure, including the percentage changes in ownership for GO Central Europe, and shows the reporting currency and GL prefix for each subsidiary.


Great Outdoors Consolidated (holding company) USD
GO Americas (AMX 1099) USD
GO Asia Pacific (EAX 4199) YEN
GO Accessories (EUX 8199) EURO
Year 1 60%
Year 3 50%
Year 1 40%
Year 3 50%
GO Central Europe (CEU 6199) EURO
GO Southern Europe (SEU 7199) EURO
GO Northern Europe (NEU 5199) EURO

Each corporation has the same departmental structure and the same GL structure, shown in the table below. Divisions may not report in the same currencies. For example, the Americas subsidiary reports in US dollars, but the Corporate division local currency is Canadian dollars, and the Operations division local currency is pesos.

Division (GL)	Department (GL)
Corporate division (1700)	Sales (1720)
	Marketing (1750)
	IS&T (1760)
	Human Resources (1730)
	Finance (1740)
	Procurement (1710)
Operations (1800)	Production and Distribution (1820)
	Customer Service (1820)

Employees

The Great Outdoors data contains a full list of employees in all divisions, departments, and locations. Data is available for reports about bonuses (Global Bonus report) and sales commissions (Sales Commissions for Central Europe report), training (Employee Training by Year report), and performance reviews and employee satisfaction surveys (Employee Satisfaction 2006). If you use Metric Studio, sample metrics for human resources are also available.

In the GO Data Warehouse (analysis) package, groups of measures and the related dimensions are organized into folders. The employees are organized in hierarchies for region, manager, position or department, and organization, to make different kinds of aggregation easy to report on. Aggregation has been set for the Employee Position Summary measures, so that Position count and Planned position count aggregate to a maximum for time dimensions, and a sum for other dimensions. For example, see the Planned Headcount report.

The employees are also listed in a sample LDIF file (p. 332). This authentication directory is necessary for the Transformer 8 cubes and for Cognos Planning samples. No other samples depend on security profiles. For more information, see the *Installation and Configuration Guide*.

Sales and Marketing

Data about sales and marketing is available for all of the companies in the Great Outdoors group. GO Accessories has richer details to support analysis examples. For example, see the Revenue vs % Gross Profit by Product Brand analysis, based on the Sales and Marketing cube. Marketing and sales campaigns are tied to the Great Outdoors regional companies.

Overall, the GO companies have experienced solid growth across most product lines (Sales Growth Year Over Year), in all regions (Revenue by GO Subsidiary 2005), because of factors like an increase in repeat business and new or improved products, such as the high margin sunglasses product line. In the product lines sold by the five regional companies (all but GO Accessories) promotions have had mixed success (Promotion Success by Campaign, Bundle and Quarter). If you use Metric Studio, this can also be seen in the sample metrics.

Customer Surveys

The data also contains information from customer surveys. For example, the product line that includes bug spray, sun screen, and so on has not been successful (Product Satisfaction - Outdoor Protection 2005) and a source of retailer dissatisfaction may be the level of customer service rather than the returns (Customer Returns and Satisfaction). If you use Metric Studio, this information can also be monitored in metrics.

Sales Outlets

Revenue from the corporate outlets is available at the transaction level. Revenue from the franchise outlets is available at the consolidated level only (Sales and Marketing cube). Metrics about retailers show that the number of new retail outlets has dropped over the time period covered by this data.

GO Accessories sells worldwide, and sells only accessories. The other five subsidiaries in the group of companies are regional and sell all product lines for retailers in their region. For example, the report Top 10 Retailers in 2005 uses sparklines and list data to review revenues at the retailer level.

Great Outdoors Database, Models, and Packages

The Great Outdoors models illustrate modeling techniques and support the samples. The models are based on the GO data warehouse and the GO sales transactional database and are the basis for the sample reports and queries. Each model contains two packages for publishing analysis (dimensional) and query views of the data.

For a description of each sample report or query, see the user guide for the studio that you open the sample in. For more information about modeling techniques, see the *Guidelines for Modeling Metadata*, or the Framework Manager *User Guide*.

You must have access to Framework Manager, the modeling tool in Cognos 8, to look at the sample models. You may also need to set up the sample databases and connections. For instructions, see the *Administration and Security Guide* or the *Installation and Configuration Guide*.

GO Data Warehouse

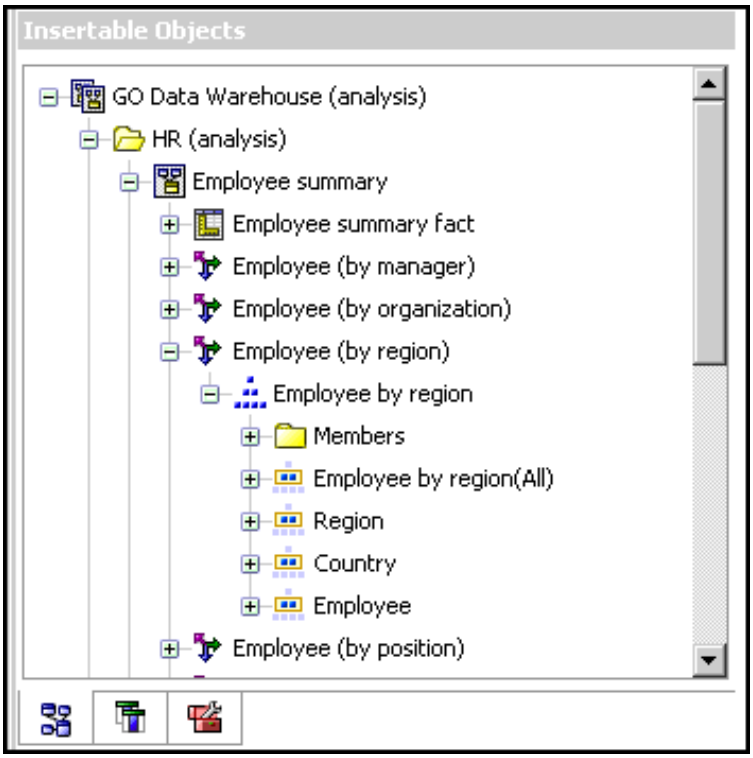
The GO Data Warehouse model, great_outdoors_data_warehouse.cpf, is based on the database GOSALESDW. It contains data about human resources, sales and marketing, and finance, grouped into business areas. In the Database view, the three business areas are grouped into separate namespaces. The Database view contains a fourth namespace (GO Data) for the common information.

The Database view is very similar to the structure of the underlying database. All tables (database query subjects) are unchanged. This enables Cognos 8 to retrieve metadata directly from the package in most cases, instead of using a metadata call to the database. The following changes and additions have been made in the Database view:

- Joins have been added as necessary.
- To allow for aggregation at different levels of granularity, some model query subjects have been created. For example, see the relationships between Time and Sales or Sales fact.
- To allow single joins to be made between the lookup tables and each level in a dimension, lookup tables have been copied. For example, see the Products look up tables.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:

- Calculations were added to the model query subjects. For example, the time dimension contains language calculations.
- Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy. For example, see the employee hierarchies, where employees are organized by manager, organization, region, and position.



The GO Sales Transactional Database

The GO Sales model, great_outdoors_sales.cpf, is based on the GOSALES database, which is structured as a transactional database. It contains principally sales data.

The Database view is very similar to the underlying database structure. The following changes and additions have been made in the Database view:

- To make it possible to join the fact tables to the time dimension, model query subjects and multipart joins have been used.
- Other joins have been added as necessary.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:

- Calculations were added to the model query subjects.
- Model query subjects that were created in the Database view to enable joins on the time dimension have been linked as reference shortcuts.
- Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy.
- Slowly changing dimensions require calculations to ensure that the correct record for the time period is retrieved. This is done with a calculation such as the one shown below, for the relationship between Sales Staff and Orders.

```
[Business view].[Sales
Staff].[Sales staff code]=[Business view].[Sales].[Sales staff code]
and ([Business view].[Sales].[Day key (order date)] between [Business
view].[Sales Staff].[Day key (record start date)]  and [Business
view].[Sales Staff].[Day key (period record end date)] )
```

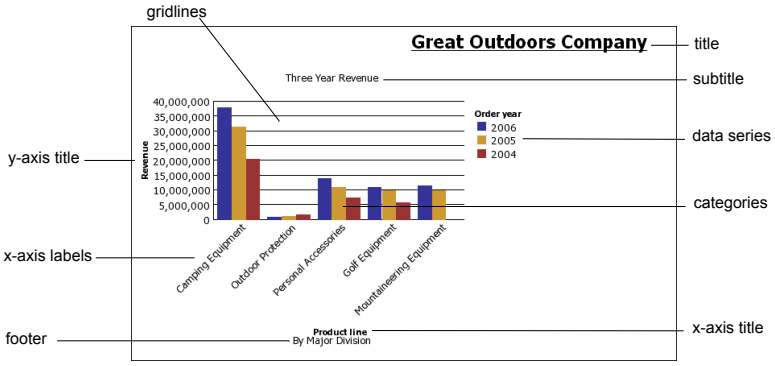
Appendix C: Chart Types

Charts are a graphical way of presenting data. For example, you can create a chart to visualize how actual sales compare to projected sales, or to discover whether sales are falling or rising over quarterly periods.

Some chart types are not supported for Microsoft Excel output, or appear differently. For more information, see "Producing Reports in Microsoft Excel Format" (p. 515)

Chart Elements

This column chart shows the most common chart elements. You can add extra elements such as baselines and notes using Report Studio.



Axes

Axes are lines that provide a frame of reference for measurement or comparison.

The y-axis refers to measures of quantitative data, such as sales figures or quantities. Charts may have more than one y-axis.

The x-axis or ordinal axis plots qualitative data, such as products or regions. It runs horizontally, except in bar charts.

The z-axis is the vertical axis in a 3-D chart.

Gridlines

Axes are lines that provide a frame of reference for measurement or comparison. Major gridlines extend from the tick marks on an axis and run behind the data markers.

Data Series

A data series is a group of related data points plotted in a chart. Each series has a unique color or pattern and is described in the legend.

In the example chart, the data series are order years 2004, 2005, and 2006.

Legend

A legend is a key to the patterns or colors assigned to the data series or categories in a chart.

Categories

Categories are groups of related data from the data series, plotted on the x-axis. Categories of multiple data series are shown together using clustered and stacked data markers.

In the example chart, the categories are the product lines of The Great Outdoors Company, in clustered columns.

Columns, Lines, and Areas

Charts use columns, lines, and areas as visual representations of data points. Other examples of visual representations include horizontal bars, points, and bubbles.

Choosing a Chart Type and Configuration

To choose a chart type, consider what you want the chart to illustrate. Different chart types and configurations emphasize different things.

Purpose	Chart type or configuration
Show contributions of parts to a whole	pie stacked configuration 100% stacked configuration
Show trends in time or contrast values across different categories	line area bar column
Compare groups of related information against actual values	standard configuration radar 3D
Compare different kinds of quantitative information	column-line

Pie Charts

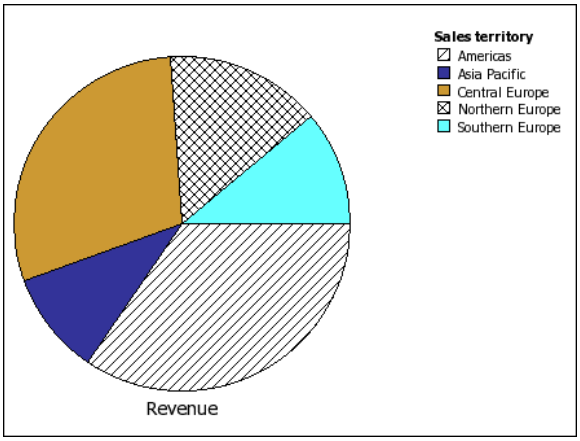
Pie charts are useful for highlighting proportions.

Pie charts use segments of a circle to show the relationship of parts to the whole. To highlight actual values, we recommend that you use another chart type, such as a stacked chart.

Pie charts plot a single data series. To avoid multiple pies when plotting multiple data series, we recommend that you use a 100% stacked chart.

Reports in PDF or HTML format are limited to show a maximum of 16 pie or gauge charts. If you need to see more, run the report in Excel Single Sheet format and they all appear in the report.

This pie chart shows that the largest proportion of revenue comes from the Americas, followed closely by the Central Europe region.



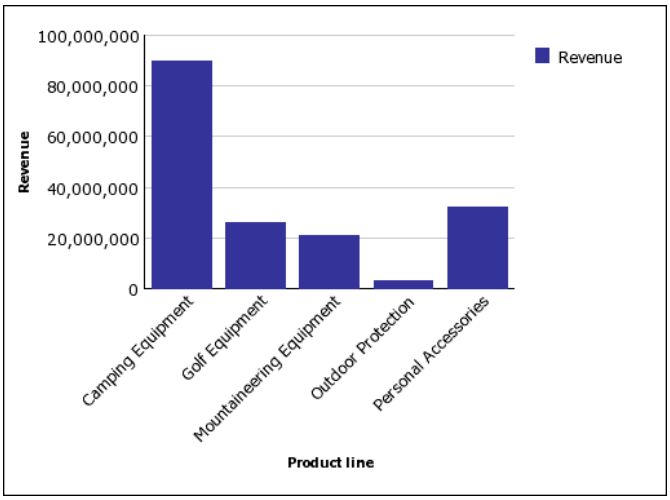
Pie charts can plot data using standard, 100%, and 3D configurations.

Column Charts

Column charts are useful to compare discrete data or to show trends over time.

Column charts use vertical data markers to compare individual values.

This column chart uses actual values to show the revenue for each product line.



Column charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

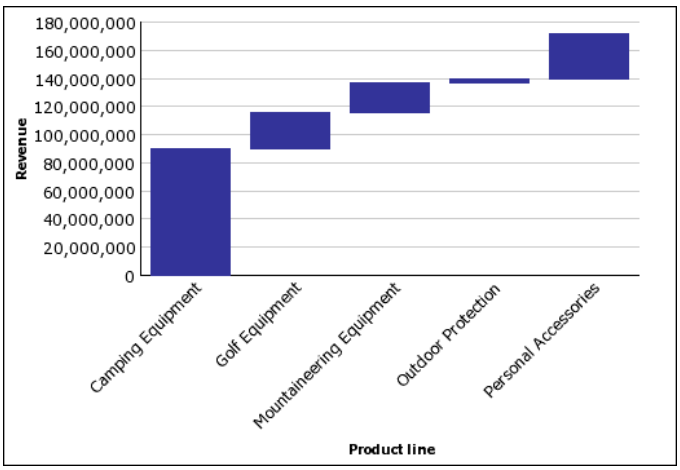
Progressive Column Charts

Progressive column charts are like stacked charts with each segment of a single stack displaced vertically from the next segment.

Progressive column charts are useful for emphasizing the contribution of the individual segments to the whole.

Progressive column charts are also known as waterfall charts. These charts are not supported for Microsoft Excel output.

This progressive column chart analyzes the contribution of each product line to revenue.



Progressive column charts can plot data using standard and 3D configurations. You can also create progressive charts using horizontal bars.

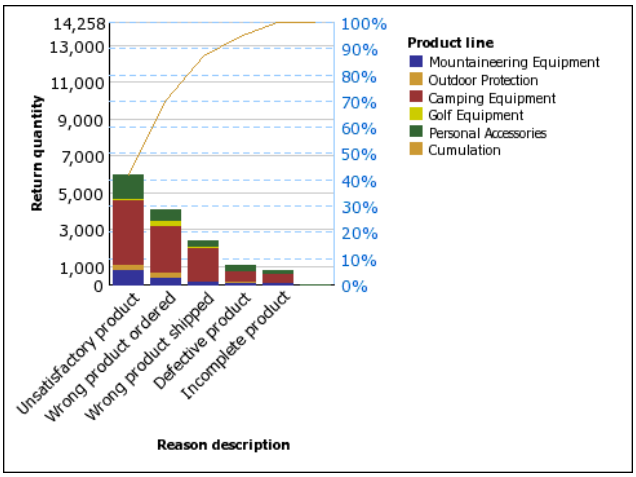
Pareto Charts

Pareto charts are useful for prioritizing and focusing process changes.

Pareto charts rank categories from the most frequent to the least frequent. It is more effective to act on the most frequent causes of events than to solve an easy yet infrequent issue.

You can create before and after comparisons of Pareto charts to show the impact of corrective actions. These charts are not supported for Microsoft Excel output.

This Pareto chart shows that the most frequent reason for product returns is unsatisfactory product.



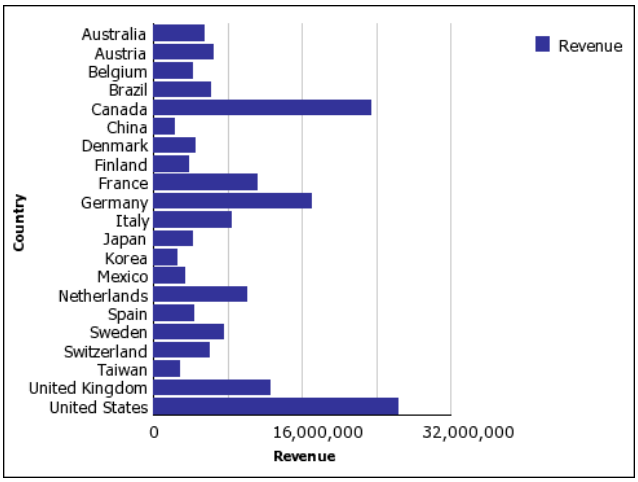
You can also create Pareto charts using horizontal bars.

Bar Charts

Bar charts are useful for showing trends over time and for charts that plot many data series.

Bar charts use horizontal data markers to compare individual values.

This bar chart shows actual revenue for every country.



Bar charts can plot data using standard, stacked, and 100% stacked configurations.

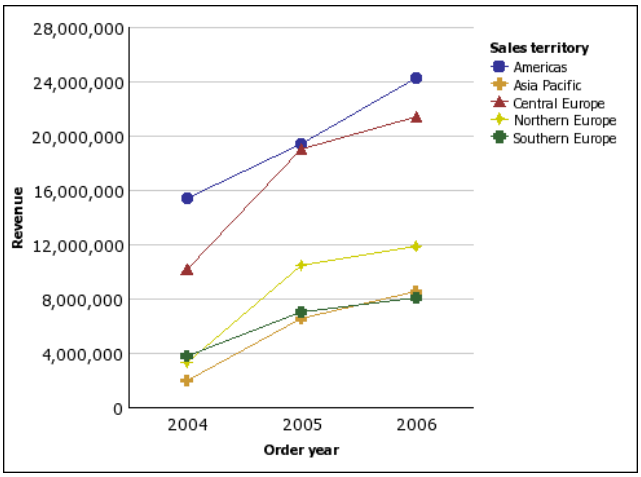
Line Charts

Line charts are useful for showing trends over time and for charts with many data series.

Line charts plot data at regular points connected by lines.

We do not recommend that you use stacked line charts because they are difficult to distinguish from unstacked line charts using multiple data series.

This line chart shows a rising revenue trend in every territory.



Line charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

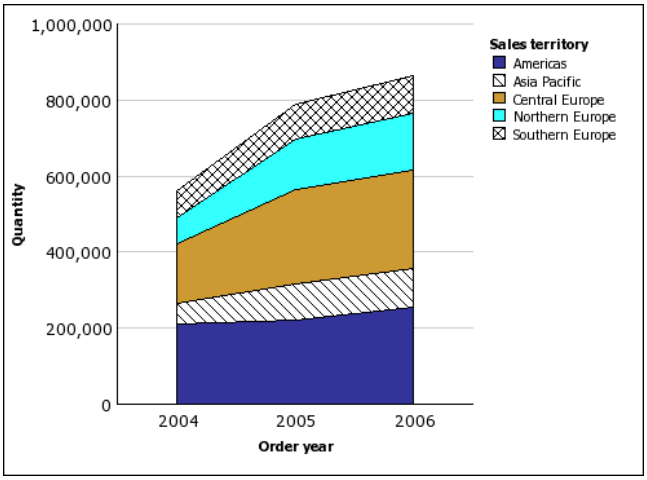
Area Charts

Area charts are useful for emphasizing the magnitude of change over time. Stacked area charts are also used to show the relationship of parts to the whole.

Area charts are like line charts that have the areas below the lines filled with colors or patterns.

We do not recommend that you use standard area charts in a chart that has multiple data series because it is possible for areas with lower values to be covered by others.

This stacked area chart shows the quantity of products sold over a two-year period in multiple territories.

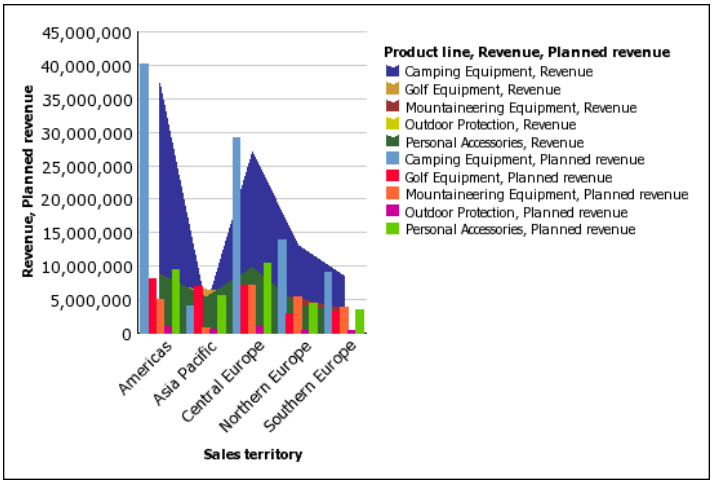


Area charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

Combination Charts

Combination charts are useful for plotting multiple data series by using combinations of columns, areas, and lines.

This combination chart shows planned revenue as a column chart and actual revenue as an area chart.



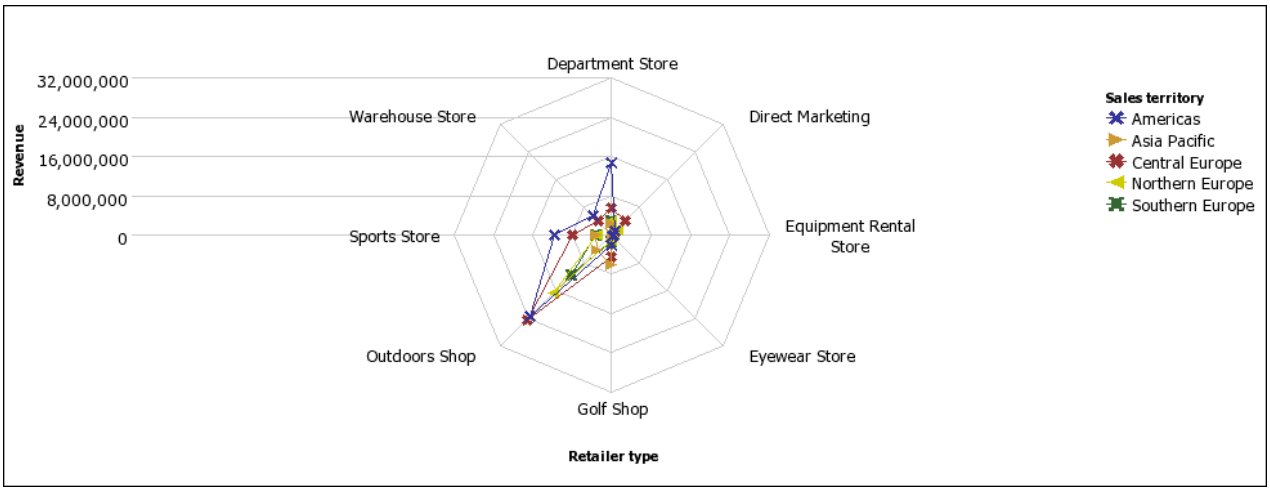
Combination charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

Radar Charts

Radar charts are useful as a comparative tool and for charts with few data series.

Radar charts integrate multiple axes into a single radial figure. Data is plotted on each axis and joined to adjacent axes by connecting lines.

This radar chart shows the revenue from multiple retailer types in multiple territories.



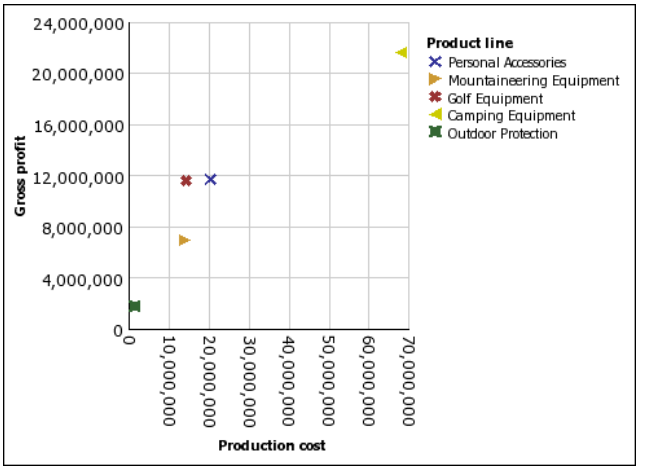
Radar charts can plot data using standard and stacked configurations.

Scatter Charts

Scatter charts use data points to plot two measures anywhere along a scale, not only at regular tick marks.

Scatter charts are useful for exploring correlations between different sets of data.

This scatter chart shows the correlation between production cost and gross profit for each product line.

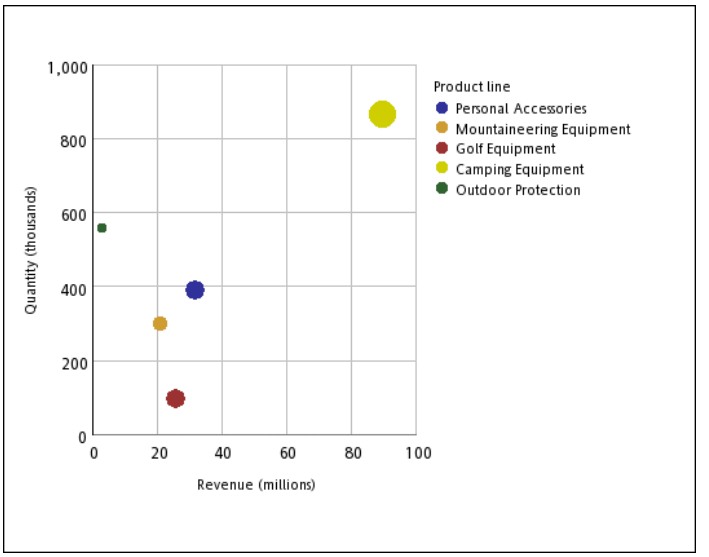


Bubble Charts

Bubble charts use data points and bubbles to plot measures anywhere along a scale, like scatter charts. The size of the bubble represents a third measure.

Bubble charts are useful for visually representing financial data. These charts are not supported for Microsoft Excel output.

This bubble chart plots quantity and revenue by product line. The size of the bubble represents gross profit.

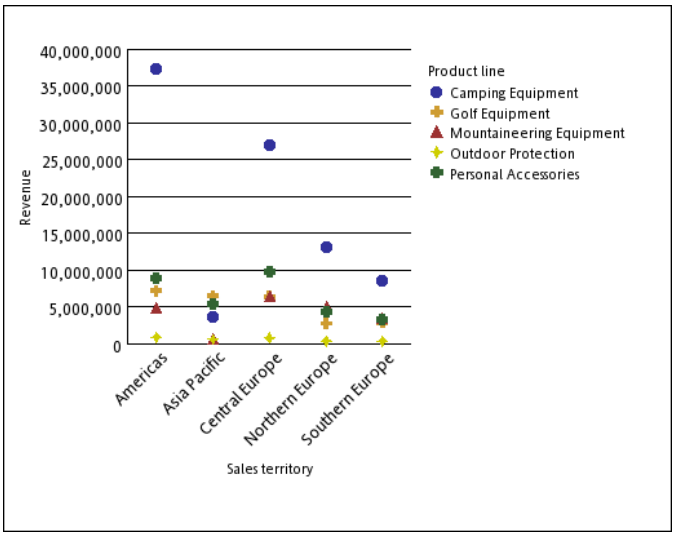


Point Charts

Point charts are useful for showing quantitative data in an uncluttered fashion.

Point charts use multiple points to plot data along an ordinal axis. A point chart is similar to a line chart without the lines. Only the data points are shown.

This point chart shows the revenue for each product line.



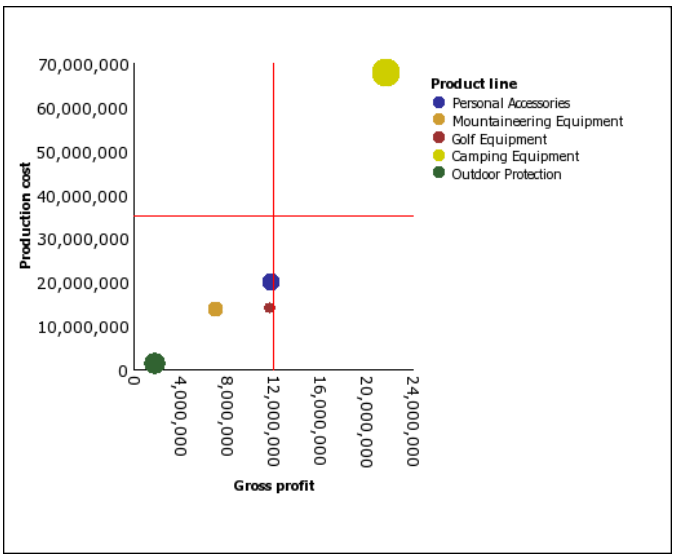
Quadrant Charts

Quadrant charts are useful for plotting data that contains three measures, using an x-axis, a y-axis, and a bubble size that represents the value of the third measure.

Quadrant charts are like bubble charts divided into four equal sections.

Use a quadrant chart to present data that can be categorized into quadrants, such as a SWOT (strengths, weaknesses, opportunities, and threats) analysis.

This quadrant chart shows the relationship between production cost and gross profit. The size of the bubble represents quantity.

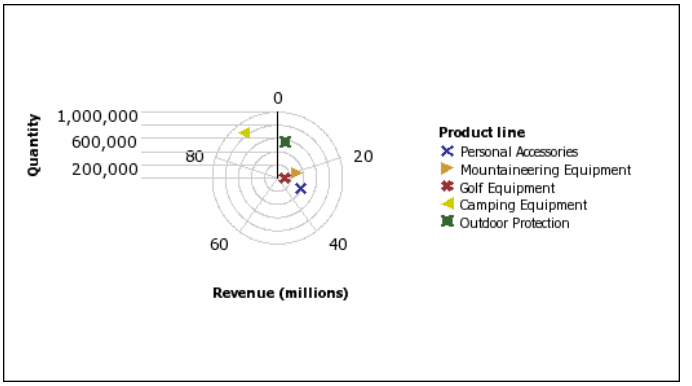


Polar Charts

Polar charts are useful for showing scientific data.

Polar charts are circular charts that use values and angles to show information as polar coordinates.

This polar chart shows the revenue and quantity for each product line. The distance along the radial axis represents revenue while the angle around the polar axis represents quantity.

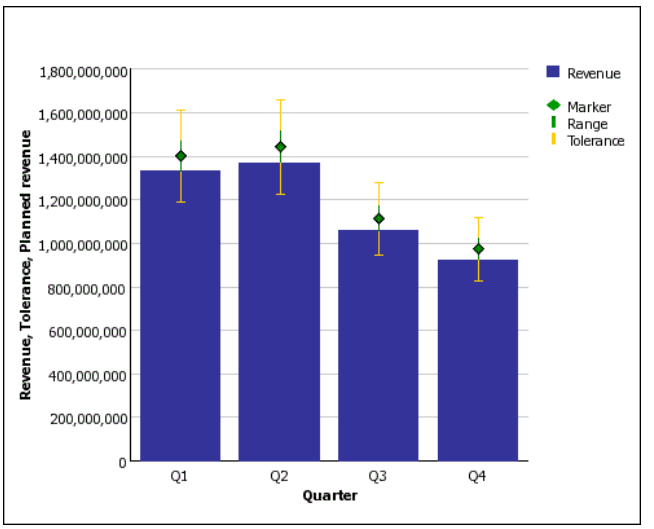


Metrics Range Charts

Metric range charts are useful for showing a target range and a tolerance range.

A metric range chart adds a target and range marker to a column, line, or area chart.

This metric range chart shows actual revenue versus planned revenue.



Gauge Charts

Gauge charts are useful for comparing values between a small number of variables either by using multiple needles on the same gauge or by using multiple gauges.

Gauge charts use needles to show information as a reading on a dial. The value for each needle is easily read against the colored data range.

Reports in PDF or HTML format are limited to show a maximum of 16 pie or gauge charts. These charts are not supported for Microsoft Excel output.

This gauge chart shows the revenue and planned revenue for each sales territory.

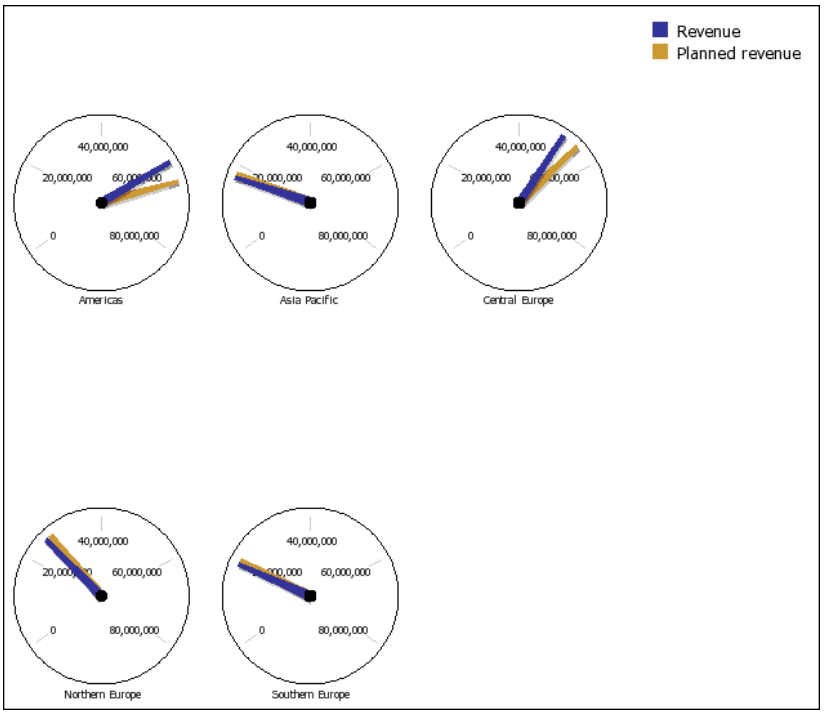


Chart Configurations

Chart configurations specify the grouping type of the columns, bars, lines, and areas in a chart. Some examples are standard, stacked, and 100% stacked charts.

Standard Charts

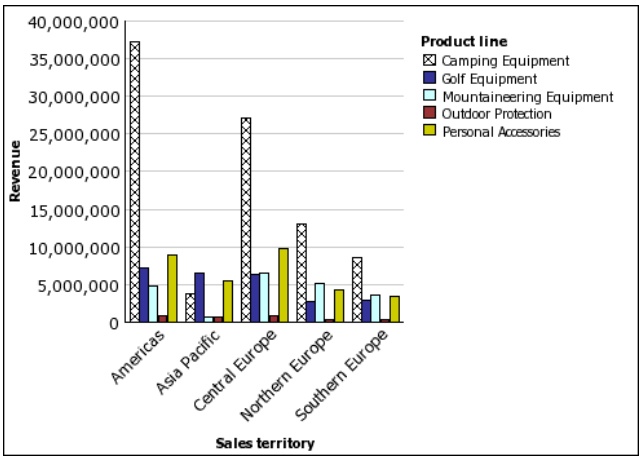
Standard - or absolute - charts are useful for comparing specific values and for representing discrete data, such as different regions or individual employees. For example, a standard column chart that plots regional sales emphasizes the actual value that each region achieves in sales.

Standard charts plot the actual value of each data series from a common axis.

When you create charts using multiple data series, you can distinguish each series by the color or pattern of its data marker. Related data series are shown together in clusters for easy comparison.

In area and radar charts that have multiple data series, areas with lower values may be covered by others.

This clustered column chart shows the revenue values for each product line within each territory.



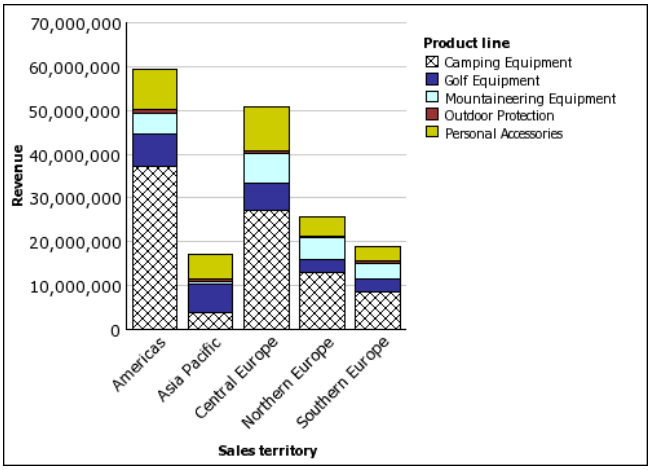
Stacked Charts

Stacked charts are useful for comparing proportional contribution within a category. They plot the relative value that each data series contributes to the total. For example, a stacked column chart that plots product line sales will emphasize the proportion that each product line contributes to the total in each territory.

You can distinguish each data series by the color or pattern of its section in the stack. The top of each stack represents the accumulated totals for each category.

We recommend that you do not use the stacked configuration in line charts that have multiple data series because it is difficult to distinguish between standard and stacked configurations.

This stacked column chart shows the high proportion that camping equipment contributed to the actual revenue in most markets.



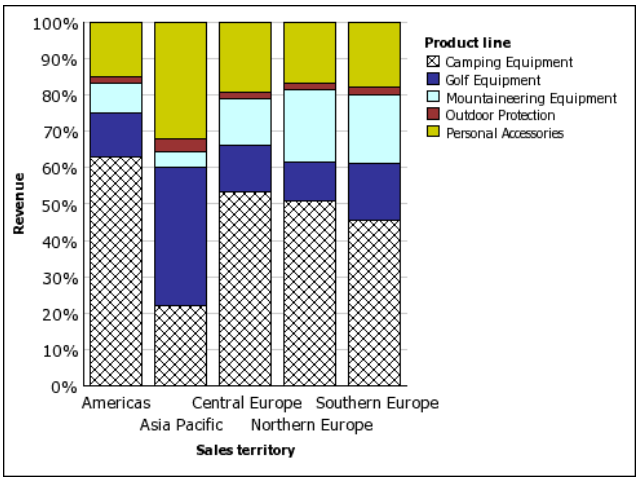
100% Stacked Charts

100% stacked charts are useful for comparing proportional contribution across all categories. They plot the relative contribution of each data series to the total, expressed as a percentage. For example, a 100% stacked column chart that plots product line sales emphasizes the percentage within each region without referring to actual values.

You can distinguish each data series by the color or pattern of its section in the stack. Each stack represents 100 per cent.

100% stacked charts highlight proportions. When actual values are important, we recommend that you use another chart configuration.

This 100% stacked chart shows the percentage of sales for each product line in each region.



3-D Charts

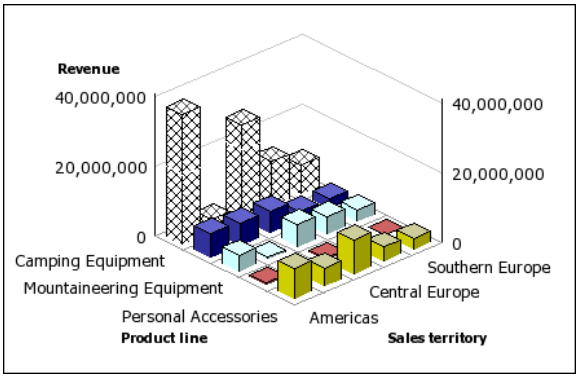
3-D charts are useful for creating charts with high visual content, such as graphics for use in presentations.

3-D column, bar, line, and area charts plot data by using three axes.

3-D pie charts have a three-dimensional visual effect.

We recommend that you do not use 3-D charts where there is a need to show exact values, such as for control or monitoring purposes. The distortion in 3-D charts can make them difficult to read accurately.

This 3-D chart shows actual revenue for each product line in each territory. Note the skipping of the labels on the x and y axes.



Appendix D: Summary Functions

This appendix describes the summary functions that are available in Report Studio. Some functions, such as Custom, are available only when you click the aggregate button on the toolbar. Additional summary functions are available in the Expression Editor (p. 357).

Aggregate

In lists, sets the summary function to **Summarize**. In crosstabs, sets the summary function to **Automatic**.

Automatic

Depending on the type of data item, applies the function **None**, **Summarize**, or **Calculated** based on the context in which the data item appears.

Calculated is applied if the underlying data source is OLAP. It is also applied if the data item expression

- contains a summary function
- is an `if-then-else` or `case` expression that contains a reference to at least a modeled measure in its condition
- contains a reference to a model calculation or to a measure that has the **Regular Aggregate** property set to a value other than **Unsupported**
- contains a reference to at least one data item that has the **Rollup Aggregation Function** property set to a value other than **None**

If the underlying data source is relational and if the data item expression contains no summary functions and a reference to at least one fact that has the **Regular Aggregate** property set to a value other than **Unsupported**, **Summarize** is applied.

If the underlying data source is OLAP and if the data item expression contains no summary functions and a reference to at least one fact that has the **Regular Aggregate** property set to a value other than **Unsupported**, **Calculated** is applied.

If the underlying data source is SAP BW, reports containing aggregations and summaries run more efficiently if the aggregation applied to a query item matches the aggregation rule for the underlying key figure on the SAP BW server. In Report Studio, the easiest way to accomplish this is to change the value of the **Aggregate Function** property to **Automatic**.

In all other contexts, **None** is applied.

For relational and dimensionally modeled relational (DMR) data sources, if this function is applied to a data item expression that has the `average` function, weighted averages are computed based on the lowest level detail values in the data source.

This is the default function.

Average

Adds all existing values and then divides by the count of existing values.

Calculated

Specifies that all the terms within the expression for a data item are aggregated according to their own rollup rules, and then the results of those aggregations are computed within the overall expression.

For example, a list contains the data item Quantity with the **Aggregation** property set to **Total**. You add a query calculation named Quantity Calculated. You define its expression as Quantity + 100 and you set its **Aggregation** property to **Calculated**. When you run the report, the values for Quantity Calculated are computed by first adding all values for Quantity and then adding 100 to each value.

In crosstabs, this function overrides any solve orders that are specified (p. 191).

Count

Counts all existing values.

If the underlying data source is OLAP, Count behaves as follows if it is specified in the **Rollup Aggregation Function** property for a data item.

Object	Behavior
Level	Count distinct is used. A warning appears when you validate the report.
Member set	Count distinct is used. A warning appears when you validate the report.
Attribute	Not supported. An error is returned when you run the report.
Measure	Supported.

If the underlying data source is dimensionally modeled relational (DMR), Count behaves as follows if it is specified in the **Aggregate Function** or **Rollup Aggregation Function** property for a data item.

Object	Behavior
Level	Count distinct is used. A warning appears when you validate the report.
Member set	Count distinct is used. A warning appears when you validate the report.
Attribute	Supported.
Measure	Supported.

Count Distinct

Returns the total number of unique non-null records.

If the underlying data source is OLAP, Count Distinct behaves as follows if it is specified in the **Rollup Aggregation Function** property for a data item.

Object	Behavior
Level	Supported.
Member set	Supported.
Attribute	Not supported. An error is returned when you run the report.
Measure	Not supported. An error is returned when you run the report.

If the underlying data source is dimensionally modeled relational (DMR), Count Distinct is supported for levels, member sets, attributes, and measures when it is specified in the **Aggregate Function** or **Rollup Aggregation Function** property for a data item.

Custom

Summarizes data based on an expression that you define.

Maximum

Selects the largest existing value.

Median

Returns the median value of the selected data item.

Minimum

Selects the smallest existing value.

None

Does not aggregate values.

If the underlying data source is relational, the data item is grouped when the query property **Auto Group & Summarize** (p. 183) is set to **Yes**.

Not Applicable

This function is available only for the **Aggregate Function** property, and it specifies that detail level aggregation does not apply. Instead, aggregates are computed based on the value of the **Rollup Aggregate Function** property.

Use this function only for reports upgraded from Cognos ReportNet(R) 1.x that contain cube data items. In Cognos ReportNet 1.x, summary functions applied to cube data items are equivalent to summary level aggregation in Cognos 8. As a result, only the summary level aggregation is applied.

Standard Deviation

Returns the standard deviation of the selected data item.

From a mathematical perspective, this function is not useful for small numbers of items, and is not supported if the query property **Auto Group & Summarize** (p. 183) is set to **Yes**.

Summarize

Aggregates data based on model or data type information. This function can be thought of as a calculate and then aggregate rule. Data is aggregated using the following rules:

If the underlying data source type is relational and the data item or calculation is a reference to a single fact query item in the model that has the **Regular Aggregate** property set to **Sum**, **Maximum**, **Minimum**, **Average**, or **Count**, aggregate data using this function. Otherwise, data is aggregated according to the data type of the data item as follows:

- **Total** is applied for numeric and interval values.
- **Maximum** is applied for date, time, and date-time values.
- **Count** is applied for everything else.

The underlying data source type can be dimensional and the data item or calculation a reference to a single item in the model. In this case, if the **Regular Aggregate** property of the model item is not **unknown**, the summary function that corresponds to the **Regular Aggregate** property is applied. If the **Regular Aggregate** property is **unknown**, set the function to **Calculated**.

For example, a list contains the data item Quantity with the **Aggregation** property set to **Average**. You add a query calculation named Quantity Summarize. You define its expression as Quantity + 100 and you set its **Aggregation** property to **Summarize**. When you run the report, the values for Quantity Summarize are computed by first adding 100 to each value for Quantity and then calculating the total. The total is calculated because Quantity contains numeric values and the expression for Quantity Summarize is not a simple data item reference. If Quantity Summarize is defined as simply Quantity, the function **Average** is applied to each value.

Total

Adds all existing values.

Variance

Returns the variance of the selected data item.

Note: From a mathematical perspective, this function is not useful for small numbers of items, and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Mapping Aggregation Functions From Data Sources to Framework Manager and Report Studio

The summary functions available in Framework Manager and Report Studio reflect summary functions supported by relational and dimensional data sources. The following table shows how summary functions in data sources are mapped to Framework Manager and Report Studio.

Data source	Framework Manager	Report Studio
None specified, or `none`	**Unsupported**	**None**
`average` (`avg`)	**Average**	**Average**
`count`	**Count**	**Count**
`count distinct`	**Count Distinct**	**Count distinct**
`maximum` (`max`)	**Maximum**	**Maximum**
`minimum` (`min`)	**Minimum**	**Minimum**
`median`	**Median**	**Median**
`standard deviation` (`stddev`)	**Standard Deviation**	**Standard Deviation**
`sum`	**Sum**	**Total**
`variance` (`var`)	**Variance**	**Variance**
`count non zero`	**Count Non-Zero**	**Automatic**
`average non zero`	**unknown**	**Automatic**
`external`	**unknown**	**Automatic**
`any`	**unknown**	**Automatic**
`calculated`	**unknown**	**Automatic**
`unknown`	**unknown**	**Automatic**
`first_period`	**unknown**	**Automatic**

Data source	Framework Manager	Report Studio
last_period	**unknown**	**Automatic**

The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

Framework Manager	Report Studio
Automatic	**Automatic**
Calculated	**Calculated**
No corresponding summary function	**Summarize**
No corresponding summary function	**Not Applicable**

Appendix E: Using the Expression Editor

An expression is any combination of operators, constants, functions, and other components that evaluates to a single value. You build expressions to create calculation and filter definitions. A calculation is an expression that you use to create a new value from existing values contained within a data item. A filter is an expression that you use to retrieve a specific subset of records.

The expression editor shows the expression components that are supported by the data source in which the metadata is stored. For example, if you import metadata from an Oracle data source, the expression editor shows only the elements that are supported in Oracle.

If you are using a DB2 data source, note that the subtract operator is invalid if you combine the datatypes `timestamp2` and `packed decimal`.

When creating an expression that will be used in a double-byte environment, such as Japanese, the only special characters that will work are ASCII-7 and ~ -- || - $ ¢ £ ¬.

Quality of Service Indicators

Not all data sources support functions the same way. The data modeler can set a quality of service indicator on functions to give a visual clue about the behavior of the functions. Report authors can use the quality of service indicators to determine which functions to use in a report. The quality of service indicators are:

- not available (X)

  This function is not available for any data source in the package.

- limited availability (!!)

  The function is not available for some data sources in the package.

- poor performance (!)

  The function is available for all data sources in the package but may have poor performance in some data sources.

- unconstrained (no symbol)

  The function is available for all data sources.

SAP BW Support

SAP BW does not support all operators or summaries. This can be confusing if you have imported SAP BW metadata and non-SAP BW metadata into the same model.

SAP BW does not support the following operators:

- like
- lookup

SAP BW does not support the following member summaries:

- date-time
- interval
- interval month
- interval day
- interval day to hour
- interval day to minute
- interval day to second
- interval hour
- interval hour to minute
- interval hour to second
- interval minute
- interval minute to second
- interval second
- interval year
- interval year to month
- moving
- running
- time with time zone
- timestamp with time zone

Cell values are date, number, or time. Attribute values are strings.

Creating Expressions Using SAP BW Data Sources

You must consider the following when creating expressions using an SAP BW data source, or you may not get the results you expect:

- The case and if/then/else constructs are not supported in calculations and filters.
- The query item identifier of the leaf-level of the 0CALDAY characteristic and its presentation hierarchies is of type date. When the values for the query item identifier are presented in Report Studio, they are formatted as dates. These formatted values should not be used in filter expressions. The correct date constant format for use in expressions is YYYY-MM-DD.
- You can apply a comparison expression with an operator other than equals to a query item that represents a level identifier. However, level identifiers are more efficient for identifying

specific values. Range comparisons must be performed on the Cognos 8 application server, which slows down the performance of the report.

Browse the Data of a Data Item

When building expressions in the expression editor, you can browse the data of a data item. This is useful when you do not know how a particular value is stored in the database. For example, you want to filter a report so that it shows data for only China. The actual value in the database for China is The People's Republic of China, and this is what you must insert in the filter expression to obtain the correct results.

Steps

1. In the source, data items, or query tab, right-click the data item you want to browse.
2. If you want to insert a single value, click **Select Value.**
3. If you want to insert multiple values, click **Select Multiple Values** .
4. In the **Values** box, click the value you want to insert in the expression.

   Tip: Use the **Words** box to search for specific values. If the data source is case sensitive and you want to perform a case insensitive search, click the **Search** arrow and then click **Case Insensitive**.
5. If you clicked multiple values, click the right arrow button to add them to the **Selected values** box.
6. Click **Insert**.

   Tip: To control where values appear in the **Expression Definition** box, you can click the copy button if you selected a single value or the copy all button if you selected multiple values instead. The values are copied to the clipboard, and you can then paste them where you want.

Searching for Values May Return Unexpected Results

In the expression editor, when searching for values for a data item, the results you obtain may contain unexpected results if the data item is not a string data type. Because users can edit the expression for a data item, Cognos 8 cannot determine with certainty what the data type is.

Therefore, Cognos 8 guesses the data type of the data item by looking at its aggregate and roll-up aggregate set.

Using Quotation Marks in Literal Strings

When inserting literal strings in an expression, you must enclose the string in single quotation marks. If the string contains a quotation mark, it must be escaped. For example, if you want to insert the string ab'c, you must type 'ab''c'.

Recommendation - Use Member Unique Name (MUN) Aliases

If you are working with a dimensional data source, use MUN aliases to simplify building reports and expressions. In addition, you can tie the MUN back to the member in the package.

When working with dimensional data sources, Cognos 8 uses MUNs to uniquely identify members. MUNs are very complex. For example, the MUN for the member 2004 might appear as follows:

```
[great_outdoors_company].[Years].[Years].[Year]->:[PC].[@MEMBER].
[20040101-20041231]
```

When you are building expressions that involve members, the use of MUNs makes these expressions difficult to read and maintain.

Tip: To view the MUN of a member, in the **Insertable Objects** pane, on the **Source** tab , right-click the member and click **Properties**.

Report Studio has an option that automatically generates an alias for MUNs (p. 57). The MUN alias is a data item that has the MUN as its definition. For the previous example, you would see 2004 as the data item, and the MUN would appear in the data item's **Expression** property. Enable this option to produce a MUN alias whenever you add a member to a report or expression.

Note: Do not modify or rename MUN aliases, because they will lose their connection to their respective members in the package.

Running a Report That Contains Members

If you are using members in a report and your modeler has updated the data source in a way that changes member unique names, the following problems may arise when you run the report.

- If the report queries an OLAP data source, you receive an error message explaining that some specific members cannot be found.
- If the report queries a dimensionally modeled relational (DMR) data source, data items whose member unique names have changed do not appear in the report. Calculations that refer to members that have changed no longer contain values from the members.

To resolve these problems, you must update the member unique names in the report. Open the query that contains the members that you need to update in Query Explorer. Delete the member from the pane in which it appears and reinsert it from the **Source** tab. For example, if you inserted the member as a detail filter, delete it from the **Detail Filters** pane and reinsert it.

Functions Not Available When Creating a Report or Layout Expression

When you create a report expression or a calculation based on a layout expression, unsupported functions do not appear in the **Functions** tab of the expression editor. Specifically, there is no **Summaries** folder, and some operators, constants, and constructs are also unavailable. These functions are not available because only the database can perform them. Report expressions and calculations based on layout expressions are performed in Report Studio.

To see the complete list of functions available in the expression editor, except for report functions, create a detail or group calculation. All functions are available when you create a detail or group calculation because these calculations are performed in the database and not in Report Studio.

Concatenating Strings

When Cognos 8 concatenates strings locally and if any of the involved strings contain null values, the result of the concatenation is an empty cell or a null value. This occurs because Cognos 8 requires that an expression that involves a null value returns a null value. Many databases ignore null strings when they perform concatenations. For example, if you concatenate strings A, B, and C, and if string B is a null value, the database may concatenate only strings A and C.

Dimensional Coercion Rules

To simplify building expressions, Cognos 8 provides coercion rules that automatically convert dimensional types. With coercion rules, you can build simpler expressions, making them easier to understand. In addition to the implicit rules that Cognos 8 provides, you can specify these rules explicitly by using various functions.

Cognos 8 supports the following types of coercion:

- coercion of an object of one dimensional type into another, such as a level into a member set
- coercion of a dimensional object into a numeric, date, time, or string value, such as a numeric measure into a numeric value

Coercion rules are applied to expressions and data items. In expressions, an operator or function may require operands to be of a particular dimensional type. When an operand is not of the required type, one or more coercion rules may be applied to coerce the operand to the appropriate type. Coercion rules may also be applied to data items to coerce the data item to a collection of members or values.

Function Operands

The following describes how coercion rules are applied to function operands:

- If the operand is of the required type, no coercion is required.
- If the function operand is supposed to be a numeric value, it is coerced into one.

  A coercion rule exists for each type of dimensional object coercion (p. 363).
- If the function operand is supposed to be a dimensional object and a coercion exists to coerce the operand to the required type, the coercion is applied.
- If no coercion exists, an error message with the code QE-DEF-0478 appears to indicate that an unsupported conversion of a dimensional object from the source to target type occurred.

Comparison and Other Symmetric Operators

Binary operators that accept operands of more than one type, such as equals (=) and `in`, need both operands to be the same dimensional type.

No coercion is possible between value domains (numeric, date, time, and string) or between members and values. Consequently, if either operand is a value type, both must be in the same value domain. Otherwise, the query fails.

Members and member set operands are valid only with the operators equals (=), not equals (<>), `in`, and `not in`, where the right side of the expression is a member, member set, or prompt. Only the following are valid:

- `[member/member set] = [member]`
- `[member/member set] <> [member]`
- `[member/member set] = ?p?`
- `[member/member set] <> ?p?`
- `[member/member set] in ([member], ...)`
- `[member/member set] not in ([member], ...)`
- `[member/member set] in ([member set])`
- `[member/member set] not in ([member set])`
- `[member/member set] in ?p?`
- `[member/member set] not in ?p?`

Exceptions

For the left operand, member sets are supported in detail and summary filter expressions, but not in expressions that use the `filter` function. Members are not supported in detail and summary filters, but they may be used in expressions that use the `filter` function.

The `in_range` operator is not supported for members and member sets.

Normal coercion rules apply to these operands.

NULL operands are considered values, rather than members.

Examples

The following examples illustrate how coercion is applied to levels in expressions with operators.

```
[Sales].[Products].[].[Product Line] = [Sales].[Products].[].[Product Line]->
[Outdoor Equipment]
```

The left operand is coerced to the following member set:

```
members( [Sales].[Products].[].[Product Line])
```

The following expressions are invalid:

- `[Sales].[Products].[].[Product Line] = NULL`
- `[Sales].[Products].[].[Product Line] + 1`

- `[Sales].[Products].[].[Product Line] = 'Canada'`
- `[Sales].[Products].[].[Product Line] > 2000`

The following examples illustrate how coercion is applied to members in expressions with operators. In each of the examples below, the left operand is coerced to a value of the current cell at this member:

```
(tuple([Sales].[Products].[].[Product Line]->[Outdoor Equipment]))
```

- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] = NULL`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + 1`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] > 2000`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + 'Canada'`

Coercion Rules for Dimensional Objects

The following table describes coercion rules that are applied when coercing an object of one dimensional type into another.

Note: Measures and Member Unique Names (MUNs) are considered members.

Source type	Target type	Result
Measure dimension	Hierarchy	A measure dimension can be used without any conversion as a hierarchy.
Measure dimension	Member set	The member set contains the members of the measure dimension. This coercion is done in two steps. First, measure dimensions are coerced to hierarchies, and then hierarchies are coerced to member sets.
Hierarchy	Member set	A set containing the members of the hierarchy.
Level	Member set	A set containing the members of the level. The equivalent expression is `members(level)`.
Measure	Member	A measure can be used without any conversion as a member.
Measure	Tuple	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to tuples.

Source type	Target type	Result
Measure	Member set	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to member sets.
Member	Tuple	The tuple consists of the member. The equivalent expression is `tuple(member)`
Member	Member set	The set consists of the member. This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to member sets.

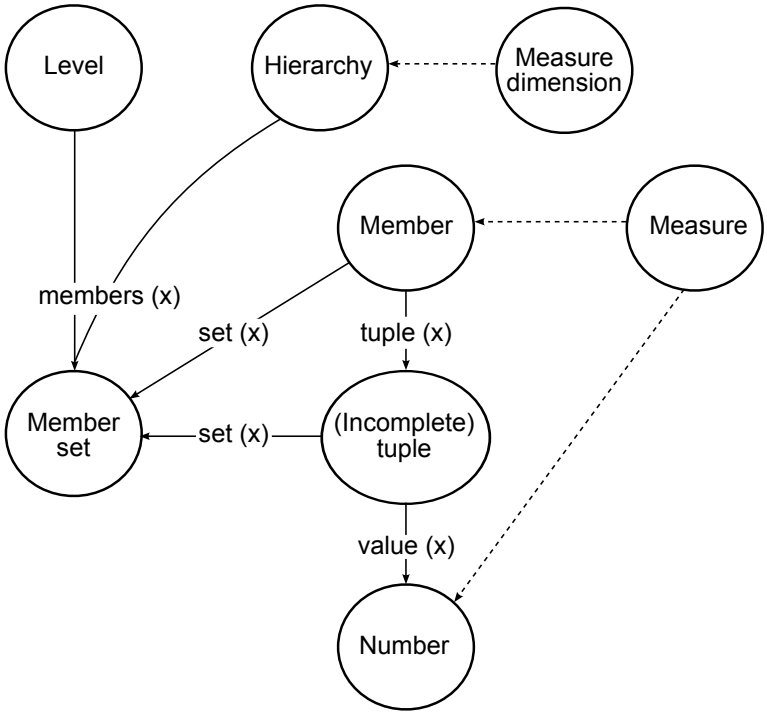
Coercion Rules for Numeric and String Values

The following table describes the coercion rules for expressions or prompts that return a numeric or string value. These rules apply to functions and operators that require an operand of a particular data type, as well as to binary operators that require operands of the same type.

Source type	Target type	Result
Measure	Number, Date/Time, or String	The result is the value of the measure at the current cube intersection.
Member	Number	The result is the value of the cell at the cube intersection specified by the member. This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to the target type.
Tuple	Number	The value of the cell at the cube intersection specified by the tuple. The equivalent expression is `value(operand or parameter)`

Coercion Paths

The following diagram shows how coercion rules are connected. The expression that coerces a source node type to a target node type is built by following the path that connects them. A dotted line indicates a coercion that does not require an explicit conversion function. This occurs whenever the source type is a specialization of the target type. For example, every measure is a member so you can use measures without conversion wherever a member is required.



Operators

Operators specify what happens to the values on either side of the operator. Operators are similar to functions, in that they manipulate data items and return a result.

(

Inserts an open parenthesis in your expression.

Syntax

```
(expression)
```

)

Inserts a close parenthesis in your expression.

Syntax

```
(expression)
```

*

Multiplies two numeric values.

Syntax

```
value1 * value2
```

,

Separates expression components.

Syntax

```
expression(param1, param2)
```

/

Divides two numeric values.

Syntax

```
value1 / value2
```

||

Concatenates strings.

Syntax

```
string1 || string2
```

+

Adds two values.

Syntax

```
value1 + value2
```

-

Subtracts two numeric values or negates a numeric value.

Syntax

```
value1 - value2
or
- value
```

<

Compares values against a defined value and retrieves the values that are less than the defined value.

Syntax

```
value1 < value2
```

<=

Compares values against a defined value and retrieves the values that are less than or equal to the defined value.

Syntax

```
value1 <= value2
```

<>

Compares values against a defined value and retrieves the values that are not equal to the defined value.

Syntax

```
value1 <> value2
```

=

Compares values against a defined value and retrieves the values that are equal to the defined value.

Syntax

```
value1 = value2
```

>

Compares values against a defined value and retrieves the values that are greater than the defined value.

Syntax

```
value1 > value2
```

->

Used as a separator in a literal member expression.

Syntax

```
[namespace].[dimension].[hierarchy].[level]->[L1]
```

>=

Compares values against a defined value and retrieves the values that are greater than or equal to the defined value.

Syntax

```
value1 >= value2
```

and

Returns true if the conditions on both sides of the expression are true.

Syntax

```
arg1 AND arg2
```

auto

Use with summary expressions to define the scope to be adjusted based on the grouping columns in the query. The scope is context dependent.

Syntax

```
aggregate_function ( expression AUTO )
```

between

Determines if a value falls in a given range.

Syntax

```
name BETWEEN value1 and value2
```

case

Use with When, Then, Else, and End.

Syntax

```
CASE expression { WHEN expression THEN expression } [ ELSE expression ] END
```

contains

Determines if a string contains another string.

Syntax

```
string1 CONTAINS string2
```

currentMeasure

Keyword that can be used as the first argument of member summary functions.

Syntax

```
aggregate_function( currentMeasure within set expression )
```

default

Use with LOOKUP construct.

Syntax

```
lookup(....) in (....) DEFAULT(....)
```

distinct

A keyword used in an aggregate expression, to include only distinct occurrences of values. See also the function unique.

Syntax

```
DISTINCT dataItem
```

Example

```
count ( DISTINCT [OrderDetailQuantity] )
```

Result: 1704

else

Use with If or Case constructs.

Syntax

```
IF (condition) THEN .... ELSE (expression) , or CASE .... ELSE expression END
```

end

Use with Case When construct.

Syntax

```
CASE .... END
```

ends with

Determines if a string ends with a given string.

Syntax

```
string1 ENDS WITH string2
```

for

Use with summary expressions to define the scope of the aggregation in the query.

Syntax

```
aggregate_function ( expression FOR expression { , expression } )
```

for all

Use with summary expressions to define the scope to be all the specified grouping columns in the query. See also FOR clause.

Syntax

```
aggregate_function ( expression FOR ALL expression { , expression } )
```

for any

Use with summary expressions to define the scope to be adjusted based on a subset of the grouping columns in the query. Equivalent to FOR clause.

Syntax

```
aggregate_function ( expression FOR ANY expression { , expression } )
```

for report

Use with summary expressions to define the scope to be the whole query. See also FOR clause.

Syntax

```
aggregate_function ( expression FOR REPORT )
```

if

Use with Then and Else.

Syntax

```
IF (condition is true) THEN (action) ELSE (alternate action)
```

in

Determines if a value exists in a given list of values.

Syntax

```
exp1 IN (exp_list)
```

in_range

Determines if an item exists in a given list of constant values or ranges.

Syntax

```
expression IN_RANGE { constant : constant [ , constant : constant ] }
```

is missing

Determines if a value is undefined in the data.

Syntax

```
value IS MISSING
```

is null

Determines if a value is undefined in the data.

Syntax

```
value IS NULL
```

is not missing

Determines if a value is defined in the data.

Syntax

```
value IS NOT MISSING
```

is not null

Determines if a value is defined in the data.

Syntax

```
value IS NOT NULL
```

like

Determines if a string matches the pattern of another string.

Syntax

```
string1 LIKE string2
```

lookup

Finds and replaces data with a value you specify. It is preferable to use the CASE construct.

Syntax

```
LOOKUP (name) in (value1 --> value2) default (expression)
```

Example

```
lookup([Country]) in ('Canada'-->([List Price] * 0.60), 'Australia'-->([List
Price] * 0.80)) default([List Price])
```

not

Returns true if the condition is false, otherwise returns false.

Syntax

```
NOT arg
```

or

Returns true if either of the two conditions on both sides of the expression is true.

Syntax

```
arg1 OR arg2
```

prefilter

Performs a summary calculation before applying the summary filter.

Syntax

```
summary ([expression] PREFILTER)
```

rows

Counts the number of rows output by the query. Use with Count().

Syntax

```
count(ROWS)
```

starts with

Determines if a string starts with a given string.

Syntax

```
string1 STARTS WITH string2
```

then

Use with If or Case constructs.

Syntax

```
IF (condition) THEN ...., or CASE expression WHEN expression THEN .... END
```

when

Use with Case construct.

Syntax

```
CASE [expression] WHEN .... END
```

Summaries

This list contains predefined functions that return either a single summary value for a group of related values or a different summary value for each instance of a group of related values.

aggregate

Returns a calculated value using the appropriate aggregation function, based on the aggregation type of the expression.

Syntax

```
aggregate ( expr [ auto ] )
aggregate ( expr for [ all | any ] expr { , expr } )
aggregate ( expr for report )
```

average

Returns the average value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
average ( [ distinct ] expr [ auto ] )
average ( [ distinct ] expr for [ all | any ] expr { , expr } )
average ( [ distinct ] expr for report )
```

Example

```
average ( Sales )
```

Result: The average of all Sales values.

count

Returns the number of selected data items excluding NULL values. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
count ( [ distinct ] expr [ auto ] )
count ( [ distinct ] expr for [ all | any ] expr { , expr } )
count ( [ distinct ] expr for report )
```

Example

```
count ( Sales )
```

Result: The total number of entries under Sales.

maximum

Returns the maximum value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
maximum ( [ distinct ] expr [ auto ] )
maximum ( [ distinct ] expr for [ all | any ] expr { , expr } )
maximum ( [ distinct ] expr for report )
```

Example

```
maximum ( Sales )
```

Result: The maximum value of all Sales values.

median

Returns the median value of selected data items.

Syntax

```
median ( expr [ auto ] )
median ( expr for [ all | any ] expr { , expr } )
median ( expr for report )
```

minimum

Returns the minimum value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
minimum ( [ distinct ] expr [ auto ] )
minimum ( [ distinct ] expr for [ all | any ] expr { , expr } )
minimum ( [ distinct ] expr for report )
```

Example

```
minimum ( Sales )
```

Result: The minimum value of all Sales values.

moving-average

Returns a moving average by row for a specified set of values of over a specified number of rows. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
moving-average ( numeric_expr , numeric_expr  [ at exp {, expr } ]  [
<for-option> ] [ prefilter ] )
moving-average ( [ distinct ] numeric_expr , numeric_expr [ <for-option> ] [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
moving-average ( Qty, 3 )
```

Result: For each row, this displays the quantity and a moving average of the current row and the preceding two rows.

Qty	Moving-Average (Qty, 3)
200	200
700	450
400	433.3333
200	433.3333
200	266.6667
500	300.0000

moving-total

Returns a moving total by row for a specified set of values over a specified number of rows. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
moving-total ( numeric_expr , numeric_expr  [ at exp {, expr } ]  [ <for-option>
 ] [ prefilter ] )
moving-total ( [ distinct ] numeric_expr , numeric_expr [ <for-option> ] [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
moving-total ( Quantity, 3 )
```

Result: For each row, this displays the quantity and a moving total of the current row and the preceding two rows.

Qty	Moving-Total (Qty, 3)
200	200
700	900
400	1300
200	1300
200	800
500	900

percentage

Returns the percent of the total value for selected data items. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
percentage ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [ prefilter
] )
percentage ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
percentage ( sales 98 )
```

Result: Shows the percentage of the total sales for 1998 that is attributed to each sales representative.

Sales Rep	Sales 98	Percentage
Bill Gibbons	60646	7.11%
Bjorn Flertjan	62523	7.35%
Chris Cornel	22396	2.63%

percentile

Returns a value, on a scale of one hundred, that indicates the percent of a distribution that is equal to or below the selected data items. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
percentile ( numeric_expr  [ at exp {, expr } ] [ <for-option> ] [ prefilter ]
 )
percentile ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
percentile ( Sales 98 )
```

Result: For each row, displays the percentage of rows that are equal to or less than the quantity value of that row.

Qty	Percentile (Qty)
800	1
700	0.875
600	0.75
500	0.625
400	0.5
400	0.5
200	0.25
200	0.25

quantile

Returns the rank of a value for a range that you specify. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest). The <for-option> defines the scope of the function.

The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
quantile ( numeric_expr, numeric_expr  [ at exp {, expr } ]  [ <for-option> ]
[ prefilter ] )
quantile ( [ distinct ] numeric_expr, numeric_expr  [ <for-option> ] [ prefilter
 ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
quantile ( Qty, 4 )
```

Result: The quantity, the rank of the quantity value, and the quantity values broken down into 4 quantile groups (quartiles).

Qty	Rank (Qty)	Quantile (Qty, 4)
800	1	1
700	2	1
600	3	2
500	4	2
400	5	3
400	5	3
200	7	4
200	7	4

quartile

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
quartile ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [ prefilter ]
)
quartile ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
quartile ( Qty )
```

Result: This displays the quantity and the quartile of the quantity value represented as integers from 1 (highest) to 4 (lowest).

```
Qty               Quartile (Qty)
------            ----------------------
450               1
400               1
350               2
300               2
250               3
200               3
150               4
100               4
```

rank

Returns the rank value of selected data items. The sort order is optional; descending order (DESC) is assumed by default. If two or more rows tie, then there is a gap in the sequence of ranked values (also known as Olympic ranking). The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product. Null values are ranked last.

Syntax

```
rank ( expr [ ASC | DESC ] {, expr [ ASC | DESC ] }  [ at exp {, expr } ]  [
<for-option> ] [ prefilter ] )
rank ( [ distinct ] expr [ ASC | DESC ] {, expr [ ASC | DESC ] } [ <for-option>]
  [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
rank ( Sales 98 )
```

Result: For each row, this displays the rank value of sales for 1998 that is attributed to each sales representative, and skips some numbers when there is a tie between rows.

```
Sales Rep          Sales 98     Rank
———————————        ———————————  ————————————
Bill               60000        1
Gibbons
                   50000        2
Bjorn
Flertjan           50000        2
                   48000        4
Chris
Cornel
John Smith
```

running-average

Returns the running average by row (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-average ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-average ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-average ( Qty )
```

Result: For each row, this displays the quantity and a running average of the current and the previous rows.

Name	Qty	Avg	Running-Average for name
Smith	7	5	7
Smith	3	5	5
Smith	6	5	5.33
Smith	4	5	5
Wong	3	4	3
Wong	5	4	4

running-count

Returns the running count by row (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-count ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [ prefilter
 ] )
running-count ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-count ( Qty )
```

Result: For each row, this displays the quantity and a running count of the position of current row.

```
Name        Qty         Count       Running-Count for name
-------     ------      ---------   --------------------------------------
Smith       7           4           1
Smith       3           4           2
Smith       6           4           3
Smith       4           4           4
Wong        3           3           1
Wong        5           3           2
```

running-difference

Returns a running difference by row, calculated as the difference between the value for the current row and the preceding row, (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-difference ( numeric_expr [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-difference ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ]
 )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-difference ( Qty )
```

Result: For each row, this displays the quantity and a running difference between the value for the current row and the preceding row.

```
Name        Qty         Running-Difference for name
-------     ------      ------------------------------------
Smith       7           NULL
Smith       3           -4
Smith       6           3
Smith       4           -2
Wong        3           -1
Wong        5           2
```

running-maximum

Returns the running maximum by row (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and

can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-maximum (  numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-maximum ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-maximum ( Qty )
```

Result: For each row, this displays the quantity and a running maximum of the current and previous rows.

Name	Qty	Max	Running-Maximum (Qty) for name
Smith	2	7	2
Smith	3	7	3
Smith	6	7	6
Smith	7	7	7
Wong	3	5	3
Wong	5	5	5

running-minimum

Returns the running minimum by row (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-minimum ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-minimum ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-minimum ( Qty )
```

Result: For each row, this displays the quantity and a running minimum of the current and previous rows.

Name	Qty	Min	Running-Minimum (Qty) for name
Smith	7	2	7
Smith	3	2	3
Smith	6	2	3
Smith	2	2	2
Wong	4	3	4
Wong	5	3	4

running-total

Returns a running total by row (including the current row) for a set of values. The <for-option> defines the scope of the function. The at option defines the level of aggregation and can only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-total ( numeric_expr [ at exp {, expr } ]   [ <for-option> ] [ prefilter ] )
running-total ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-total ( Qty )
```

Result: For each row, this displays the quantity and a running total of the current and previous rows.

Name	Qty	Total	Running-Total (Qty) for name
Smith	2	18	2
Smith	3	18	5
Smith	6	18	11
Smith	7	18	18
Wong	3	12	3
Wong	5	12	8

standard-deviation

Returns the standard deviation of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
standard-deviation ( [ distinct ] expr [ auto ] )
standard-deviation ( [ distinct ] expr for [ all | any ] expr { , expr } )
standard-deviation ( [ distinct ] expr for report )
```

Example

```
standard-deviation ( ProductCost )
```

Result: A value indicating the deviation between product costs and the average product cost.

standard-deviation-pop

Computes the population standard deviation and returns the square root of the population variance. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
standard-deviation-pop ( [ distinct ] expr [ auto ] )
standard-deviation-pop ( [ distinct ] expr for [ all | any ] expr { , expr } )

standard-deviation-pop ( [ distinct ] expr for report )
```

Example

```
standard-deviation-pop ( ProductCost )
```

Result: A value of the square root of the population variance.

total

Returns the total value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
total ( [ distinct ] expr [ auto ] )
total ( [ distinct ] expr for [ all | any ] expr { , expr } )
total ( [ distinct ] expr for report )
```

Example

```
total ( Sales )
```

Result: The total value of all Sales values.

variance

Returns the variance of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
variance ( [ distinct ] expr [ auto ] )
variance ( [ distinct ] expr for [ all | any ] expr { , expr } )
variance ( [ distinct ] expr for report )
```

Example

```
variance ( Product Cost )
```

Result: A value indicating how widely product costs vary from the average product cost.

variance-pop

Returns the population variance of a set of numbers after discarding the nulls in this set.

Syntax

```
variance-pop ( [ distinct ] expr [ auto ] )
variance-pop ( [ distinct ] expr for [ all | any ] expr { , expr } )
variance-pop ( [ distinct ] expr for report )
```

Example

```
variance-pop ( Qty)
```

Result: For each row, this displays the population variance of a set of numbers after discarding the nulls in this set.

Member Summaries

This list contains predefined functions that return either a single summary value for a set of members or a different summary value for each member of a set of members.

aggregate

Returns a calculated value using the appropriate aggregation function, based on the aggregation type of the expression.

Syntax

```
aggregate ( < currentMeasure | expr > within set set_expr { , set_expr } )
aggregate ( < currentMeasure | expr > within < detail | aggregate > expr )
```

average

Returns the average value of selected data items.

Syntax

```
average ( < currentMeasure | expr > within set set_expr { , set_expr } )
average ( < currentMeasure | expr > within < detail | aggregate > expr )
```

count

Returns the number of selected data items excluding NULL values.

Syntax

```
count ( < currentMeasure | expr > within set set_expr { , set_expr } )
count ( < currentMeasure | expr > within < detail | aggregate > expr )
```

maximum

Returns the maximum value of selected data items.

Syntax

```
maximum ( < currentMeasure | expr > within set set_expr { , set_expr } )
maximum ( < currentMeasure | expr > within < detail | aggregate > expr )
```

median

Returns the median value of selected data items.

Syntax

```
median ( < currentMeasure | expr > within set set_expr { , set_expr } )
median ( < currentMeasure | expr > within < detail | aggregate > expr )
```

minimum

Returns the minimum value of selected data items.

Syntax

```
minimum ( < currentMeasure | expr > within set set_expr { , set_expr } )
minimum ( < currentMeasure | expr > within < detail | aggregate > expr )
```

percentage

Returns the percent of the total value for selected data items.

Syntax

```
percentage ( numeric_expr  [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

Example

```
percentage ( [gosales].[sales measures].[quantity] tuple [gosales].[Staff].[].
[department]->[West] within set children([gosales].[Staff].[].[Staff] )
```

percentile

Returns a value, on a scale of one hundred, that indicates the percent of a distribution that is equal to or below the selected data items.

Syntax

```
percentile ( numeric_expr  [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

quantile

Returns the rank of a value for the specified range. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest).

Syntax

```
quantile ( numeric_expr, numeric_expr  [ tuple member_expr {, member_expr } ]
within set set_expr  {, set_expr } )
```

quartile

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values.

Syntax

```
quartile ( numeric_expr [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

rank

Returns the rank value of selected data items. The type of ranking returned (Olympic, dense or serial) is data source dependent. The sort order is optional; DESC is assumed by default.

Syntax

```
rank ( numeric_expr [ ASC | DESC ] [ tuple member_expr {, member_expr } ] within
 set set_expr  {, set_expr } )
```

Example

```
rank  ( [gosales].[sales measures].[quantity] tuple [gosales].[Staff].[].
[department]->[West] within set children([gosales].[Staff].[].[Staff] )
```

standard-deviation

Returns the standard deviation of selected data items.

Syntax

```
standard-deviation ( < currentMeasure | expr > within set set_expr { , set_expr
 } )
standard-deviation ( < currentMeasure | expr > within < detail | aggregate >
expr )
```

total

Returns the total value of selected data items.

Syntax

```
total ( < currentMeasure | expr > within set set_expr { , set_expr } )
total ( < currentMeasure | expr > within < detail | aggregate > expr )
```

variance

Returns the variance of selected data items.

Syntax

```
variance ( < currentMeasure | expr > within set set_expr { , set_expr } )
variance ( < currentMeasure | expr > within < detail | aggregate > expr )
```

Constants

A constant is a fixed value that you can use in an expression.

date

Inserts the current system date.

date-time

Inserts the current system date and time.

time with time zone

Inserts a zero time with time zone.

timestamp with time zone

Inserts an example of a timestamp with time zone.

interval

Inserts a zero interval.

interval year

Inserts a zero year interval.

interval month

Inserts a zero month interval.

interval year to month

Inserts a zero year to month interval.

interval day

Inserts a zero day interval.

interval hour

Inserts a zero hour interval.

interval minute

Inserts a zero minute interval.

interval second

Inserts a zero second interval.

interval day to hour

Inserts a zero day to hour interval.

interval day to minute

Inserts a zero day to minute interval.

interval day to second

Inserts a zero day to second interval.

interval hour to minute

Inserts a zero hour to minute interval.

interval hour to second

Inserts a zero hour to second interval.

interval minute to second

Inserts a zero minute to second interval.

null

Inserts a null value if the expression conditions are not met.

number

Inserts the number 0, which you can replace with a new numeric value.

string

Inserts an empty string.

time

Inserts the current system time.

Constructs

A construct is a template that can be used to create an expression.

if then else

This is the template for using an "IF...THEN...ELSE" statement in the expression.

Syntax

```
IF ([Country] = 'Canada')
    THEN ([List Price] * 0.60)
    ELSE ([List Price])
```

in_range

This is the template for using "IN_RANGE" in the expression.

Syntax

```
example: [gosales].[CONVERSIONRATE].[COUNTRYCODE] IN_RANGE { :30 , 40, 50, 999:
 }
```

search case

This is the template for using a "search case" in the expression.

Syntax

```
CASE
    WHEN [Country] = 'Canada' THEN ([List Price] * 0.60)
    WHEN [CountryCode] > 100 THEN [List Price] * 0.80
    ELSE [List Price]
END
```

simple case

This is the template for using a "simple case" in the expression.

Syntax

```
CASE [Country]
    WHEN 'Canada' THEN ([List Price] * 0.60)
    WHEN 'Australia' THEN [List Price] * 0.80
    ELSE [List Price]
END
```

Business Date/Time Functions

This list contains business functions for performing date and time calculations.

_add_days

Returns the date or datetime dependent on the first argument resulting from adding integer_exp days to date_exp.

Syntax

```
_add_days ( date_exp, integer_exp )
```

Example 1

```
_add_days ( 2002-04-30 , 1 )
```

Result: 2002-05-01

Example 2

```
_add_days ( 2002-04-30 12:10:10.000, 1 )
```

Result: 2002-05-01 12:10:10.000

Example 3

```
_add_days ( 2002-04-30 00:00:00.000, 1/24 )
Note that the second argument is not a whole number. This is supported by some
 database technologies and increments the time portion.
```

Result: 2002-04-30 01:00:00.000

_add_months

Returns the date or datetime dependent on the first argument resulting from adding integer_exp months to date_exp.

Syntax

```
_add_months ( date_exp, integer_exp )
```

Example 1

```
_add_months ( 2002-04-30 , 1 )
```

Result: 2002-05-30

Example 2

```
_add_months ( 2002-04-30 12:10:10.000, 1 )
```

Result: 2002-05-30 12:10:10.000

_add_years

Returns the date or datetime dependent on the first argument resulting from adding integer_exp years to date_exp.

Syntax

```
_add_years ( date_exp, integer_exp )
```

Example 1

```
_add_years ( 2002-04-30 , 1 )
```

Result: 2003-04-30

Example 2

```
_add_years ( 2002-04-30 12:10:10.000 , 1 )
```

Result: 2003-04-30 12:10:10.000

_age

Returns a number that is obtained from subtracting date_exp from today's date. This value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

Syntax

```
_age (date_exp )
```

Example

```
Today's date=2003-02-05 _age ( 1990-04-30 )
```

Result: 120906 that is 12 years, 9 months and 6 days

_day_of_week

Returns the day of week (between 1 and 7), where 1 is the first day of the week as indicated by the second parameter(between 1 and 7, 1 being Monday and 7 being Sunday). Note that in ISO 8601 standard, a week begins with Monday being day 1. In North America where Sunday is the first day of the week being day 7.

Syntax

```
_day_of_week ( date_exp, integer )
```

Example

```
_day_of_week ( 2003-01-01, 1 )
```

Result: 3

_day_of_year

Returns the ordinal for the day of the year in date_ exp (1 to 366). Also known as Julian day.

Syntax

```
_day_of_year ( date_exp )
```

Example

```
_day_of_year ( 2003-03-01 )
```

Result: 61

_days_between

Returns a positive or negative number representing the number of days between the two date expressions. If date_exp1 < date_exp2 then the result will be a negative number.

Syntax

```
_days_between ( date_exp1, date_exp2 )
```

Example

```
_days_between ( 2002-04-30 , 2002-06-21 )
```

Result: -52

_days_to_end_of_month

Returns a number representing the number of days remaining in the month represented by the date expression date_exp.

Syntax

```
_days_to_end_of_month ( date_exp )
```

Example

```
_days_to_end_of_month ( 2002-04-20 14:30:22.123 )
```

Result: 10

_first_of_month

Returns a date or datetime dependent on the argument obtained from converting date_exp to a date with the same year and month but the day set to 1.

Syntax

```
_first_of_month ( date_exp )
```

Example 1

```
_first_of_month ( 2002-04-20 )
```

Result: 2002-04-01

Example 2

```
_first_of_month ( 2002-04-20 12:10:10.000 )
```

Result: 2002-04-01 12:10:10.000

_last_of_month

Returns a date or datetime dependent on the argument that is the last day of the month represented by date_exp.

Syntax

```
_last_of_month ( date_exp )
```

Example 1

```
_last_of_month ( 2002-01-14 )
```

Result: 2002-01-31

Example 2

```
_last_of_month ( 2002-01-14 12:10:10.000 )
```

Result: 2002-01-31 12:10:10.000

_make_timestamp

Returns a timestamp constructed from integer_exp1 (the year), integer_exp2 (the month) and integer_exp3 (the day). The time portion defaults to 00:00:00.000 .

Syntax

```
_make_timestamp ( integer_exp1, integer_exp2, integer_exp3 )
```

Example

```
_make_timestamp ( 2002 , 01 , 14 )
```

Result: 2002-01-14 00:00:00.000

_months_between

Returns a positive or negative integer number representing the number of months between date_exp1 to date_exp2. If date_exp1 < date_exp2, then a negative number is returned.

Syntax

```
_months_between ( date_exp1, date_exp2 )
```

Example

```
_months_between ( 2002-01-30, 2002-04-03 )
```

Result: 2

_week_of_year

Returns the number of the week of the year of the date_exp according to the ISO 8601 standard, in which week 1 of the year is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on Monday (day 1) and ends on Sunday (day 7).

Syntax

```
_week_of_year ( date_exp )
```

Example

```
_week_of_year ( 2003-01-01 )
```

Result: 1

_years_between

Returns a positive or negative integer number representing the number of years from date_exp1 to date_exp2. If date_exp1 < date_exp2 then a negative value is returned.

Syntax

```
_years_between ( date_exp1, date_exp2 )
```

Example

```
_years_between ( 2003-01-30, 2001-04-03 )
```

Result: 1

_ymdint_between

Returns a number representing the difference between the date expressions date_exp1 and date_exp2. This value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

Syntax

```
_ymdint_between ( date_exp1, date_exp2 )
```

Example

```
_ymdint_between ( 1990-04-30 , 2003-02-05 )
```

Result: 120906 that is 12 years, 9 months and 6 days

Block Functions

This list contains functions used to access members of a set, usually in the context of Analysis Studio.

_firstFromSet

Returns the first members found in the set up to numeric_exp_max + numeric_exp_overflow. If numeric_exp_max + numeric_exp_overflow is exceeded, then only the max number of members are returned.

Syntax

```
_firstFromSet ( set_exp, numeric_exp_max , numeric_exp_overflow )
```

_remainderSet

The member expression will be included in the returned set when the size of the set_exp set is greater than numeric_exp.

Syntax

```
_remainderSet (member_exp,  set_exp , numeric_exp )
```

Macro Functions

This list contains functions that can be used within a macro. A macro may contain one or more macro functions. A macro is delimited by a number sign (#) at the beginning and at the end. Everything between the number signs is treated as a macro expression, which is executed at run time.

+

Concatenates two strings.

Syntax

```
value1 + value2
```

Example

```
# '{' + $runLocale + '}'#
```

Result: {en-us}

array

Constructs an array out of the list of parameters.

Syntax

```
array ( string_exp | array_exp { , string_exp | array_exp } )
```

Example

```
#csv ( 'x1' , 'x2' , array ( 'a1' , 'a2' ) )#
```

Result: 'x1' , 'x2' , 'a1' , 'a2'

csv

Constructs a comma separated values string from the elements of the array. Optionally the separator and quote strings can be specified. The default separator is a comma (,) and the default quote character is a single quote (').

Syntax

```
csv ( array_exp [ , separator_string [ , quote_string ] ] )
```

Example

```
#csv ( array ( 'a1' , 'a2' ) )#
```

Result: 'a1' , 'a2'

dq

Surround the passed string with double quotes.

Syntax

```
dq ( string_exp )
```

Example

```
#dq ( 'zero' )#
```

Result: "zero"

grep

Searches for elements of an array that match the pattern specified in the first argument. It returns an array with the elements that pass the pattern.

Syntax

```
grep ( pattern_string , array_exp )
```

Example

```
#csv ( grep ( 's' , array ( 'as', 'an', 'arts' ) ) )#
```

Result: 'as', 'arts'

join

Joins the elements of an array using the separator string.

Syntax

```
join ( separator_string , array_exp )
```

Example

```
# sq ( join ( ' | | ' , array ( 'as', 'an', 'arts' ) ) )#
```

Result: 'as | | an | | arts'

prompt

Prompt the user for a single value. Only the prompt_name argument is required. The datatype defaults to string when not specified. The prompt is optional when defaultText is specified. The text, when specified, will precede the value. A queryItem can be specified to take advantage of the Prompt Info properties of the query item. The trailing_text, when specified, will be appended to the value.

Syntax

```
prompt ( prompt_name , datatype , defaultText , text , queryItem , trailing_text
 )
```

Example

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE  > #prompt('Starting
CountryCode',
                    'integer',
                    '10'
)#
```

Result: select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > 10

promptmany

Prompt the user for one or more values. Only the prompt_name argument is required. The datatype defaults to string when not specified. The prompt is optional when defaultText is specified. The text, when specified, will precede the list of values. A queryItem can be specified to take advantage of the Prompt Info properties of the query item. The trailing_text, when specified, will be appended to the list of values.

Syntax

```
promptmany ( prompt_name , datatype , defaultText , text , queryItem ,
trailing_text )
```

Example 1

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY IN ( #promptmany ( 'CountryName'
 ) # )
```

Result: select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN ('Canada' , 'The Netherlands' , 'Russia')

Example 2

```
select . . .
from
gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL,
gosales.gosales.dbo.COUNTRY XX
where COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE
    #promptmany('Selected CountryCodes',
                    'integer',
                    ' ',
```

```
   ' and  COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (',
                  '',
                  ')'
)#
```

Result: select . . . from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL, gosales.gosales.dbo.COUNTRY XX where COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE and COUNTRY_MULTILINGUAL.COUNTRY_CODE IN ('Canada' , 'The Netherlands' , 'Russia')

sb

Surround the passed string with square brackets.

Syntax

```
sb ( string_exp )
```

Example

```
#sb ( 'abc' )#
```

Result: [abc]

sq

Surround the passed string with single quotes.

Syntax

```
sq ( string_exp )
```

Example

```
#sq ( 'zero' )#
```

Result: 'zero'

sort

Sorts the elements of the array in alphabetical order. Duplicates are retained.

Syntax

```
sort ( array_exp )
```

Example

```
#csv ( sort ( array ( 's3', 'a', 'x' ) ) )#
```

Result: 'a', 's3', 'x'

split

Splits a string or the string elements of the array into separate elements.

Syntax

```
split ( pattern_string, string_exp | array_exp )
```

Example 1

```
#csv ( split ( '::', 'ab=c::de=f::gh=i' ) )#
```

Result: 'ab=c' , 'de=f', 'gh=i'

Example 2

```
#csv ( split ( '=' , split ( '::', 'ab=c::de=f::gh=i' ) ) )#
```

Result: 'ab' , 'c' , 'de' , 'f', 'gh' , 'i'

substitute

Search for a pattern in a string or in the string elements of an array and substitute the first occurrence of the found text with other text.

Syntax

```
substitute ( pattern_string, replacement_string, string_exp | array_exp )
```

Example 1

```
#sq ( substitute ( '^cn=', '***', 'cn=help' ) )#
```

Result: '***help'

Example 2

```
#csv ( substitute ( '^cn=', '***', array ( 'cn=help' , 'acn=5' ) ) )#
```

Result: '***help' , 'acn=5'

Example 3

```
#csv ( substitute ( 'cn=', '', array ( 'cn=help' , 'acn=5' ) ) )#
```

Result: 'help' , 'a5'

unique

Removes duplicate entries from the array. The order of the elements is retained.

Syntax

```
unique ( array_exp )
```

Example

```
Example:
#csv ( unique ( array ( 's3', 'a', 's3', 'x' ) ) )#
```

Result: 's3', 'a', 'x'

urlencode

URL encodes the passed argument. Useful when specifying XML connection strings.

Syntax

```
field_one=urlencode(prompt('userValue'))
```

Example

```
urlencode(prompt('some_val'))
```

Result: `%27testValue%27`

CSVIdentityName

Use the identity information of the current authenticated user to lookup values in the specified parameter map. Each individual piece of the user's identity (account name, group names, role names) is used as a key into the map. The unique list of values that is retrieved from the map is then returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

Syntax

```
CSVIdentityName ( %parameter_map_name [ , separator_string ] )
```

Example

```
#CSVIdentityName ( %security_clearance_level_map )#
```

Result: 'level_500' , 'level_501' , 'level_700'

CSVIdentityNameList

Returns the pieces of the user's identity (account name, group names, role names) as a list of strings. The unique list of values is returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

Syntax

```
CSVIdentityNameList ( [ separator_string ] )
```

Example

```
#CSVIdentityNameList ( )#
```

Result: 'Everyone' , 'Report Administrators' , 'Query User'

CAMPassport

Returns the passport.

Syntax

```
CAMPassport ( )
```

Example

```
#CAMPassport ( )#
```

Result: 111:98812d62-4fd4-037b-4354-26414cf7ebef:3677162321

CAMIDList

Returns the pieces of the user's identity (account name, group names, role names) as a list of values separated by commas.

Syntax

```
CAMIDList ( [ separator_string ] )
```

Example

```
#CAMIDList ( )#
```

Result: CAMID("::Everyone"), CAMID(":Authors"), CAMID(":Query Users"), CAMID(":Consumers"), CAMID(":Metrics Authors")

CAMIDListForType

Returns an array of the user's identities based on the identity type (account, group, or role). It can be used with the macro functions csv or join.

Syntax

```
CAMIDListForType ( identity type )
```

Example

```
[qs].[userRole] IN ( #csv ( CAMIDListForType ( 'role' ) ) # )
```

Result: [qs].[userRole] IN ('CAMID("::System Administrators")', 'CAMID(":Authors")')

Common Functions

nullif

Returns NULL if exp1 equals exp2, otherwise returns exp1.

Syntax

```
nullif( exp1, exp2 )
```

_format

Associates a format with the expression. The keyword can be PERCENTAGE_0, PERCENTAGE_1 or PERCENTAGE_2.

Syntax

```
_format ( expr , keyword  )
```

Example

```
_format( [Unit Sale Price] / [Unit Price] , PERCENTAGE_2 )
```

Result: 0.75123 displayed as 75.12%

_round

Returns the numeric expression rounded to the integer_exp places right of the decimal point. Note: integer_exp MUST be a non-negative integer.

Syntax

```
_round ( numeric_exp, integer_exp )
```

Example

```
_round ( 1220.42369, 2 )
```

Result: 1220.42

abs

Returns the absolute value of numeric_exp. The sign of negative values is changed to positive.

Syntax

```
abs ( numeric_exp )
```

Example 1

```
abs ( 15 )
```

Result: 15

Example 2

```
abs ( -15 )
```

Result: 15

ancestor

Returns the ancestor of the specified member at either the specified (named) level or the specified number of levels above the member. Note: The result is not guaranteed to be consistent when there is more than one such ancestor.

Syntax

```
ancestor ( member, level | integer )
```

ancestors

Returns all the ancestors of a member at a specified level, or distance above the member. (Most data sources support only one ancestor at a specified level, but some support more than one. Hence the result is a member set.)

Syntax

```
ancestors ( member , level | index )
```

bottomCount

This function sorts a set according to the value of "numeric_exp" evaluated at each of the members of "set_exp", and returns the bottom "index_exp" members.

Syntax

```
bottomCount ( set_exp , index_exp , numeric_exp )
```

bottomPercent

This function is similar to bottomSum, but the threshold is "numeric_exp1" percent of the total.

Syntax

```
bottomPercent ( set_exp , numeric_exp1 , numeric_exp2 )
```

bottomSum

This function sorts on "numeric_exp2", evaluated at the corresponding member of "set_exp", and picks up the bottommost elements whose cumulative total is equal to or less than numeric_exp1.

Syntax

```
bottomSum ( set_exp , numeric_exp1 , numeric_exp2 )
```

caption

Returns the caption values of the specified argument.

Syntax

```
caption ( level | member | set_exp )
```

cast

Converts an expression to a specified data type. Some data types allow for a length and precision to be specified. Make sure that the target is of the appropriate type and size. The following datatypes can be specified: CHARACTER, VARCHAR, CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, REAL, FLOAT, DATE, TIME, TIMESTAMP, TIME WITH TIME ZONE, TIMESTAMP WITH TIME ZONE, and INTERVAL. When type casting to an INTERVAL type, one of the following interval qualifiers must be specified: YEAR, MONTH, or YEAR TO MONTH for the year-to-month interval datatype; DAY, HOUR, MINUTE, SECOND, DAY TO HOUR, DAY TO MINUTE, DAY TO SECOND, HOUR TO MINUTE, HOUR TO SECOND, or MINUTE TO SECOND for the day-to-second interval datatype. More notes for this function are included in the user guide.

Syntax

```
cast ( expression, datatype_specification )
```

Example 1

```
cast ( '123' , integer )
```

Result: 123

Example 2

```
cast ( 12345 , VARCHAR ( 10 ) )
```

Result: a string containing 12345

Notes

- When you convert a value of type TIMESTAMP to type DATE, the time portion of the timestamp value is ignored.
- When you convert a value of type TIMESTAMP to type TIME, the date portion of the timestamp is ignored.
- When you convert a value of type DATE to type TIMESTAMP, the time components of the timestamp are set to zero.
- When you convert a value of type TIME to type TIMESTAMP, the date component is set to the current system date.

- It is invalid to convert one interval datatype to the other (for instance because the number of days in a month is variable). Note that you can specify the number of digits for the leading qualifier only, i.e. YEAR(4) TO MONTH, DAY(5).
- Errors will be reported if the target type and size are not compatible with the source type and size.

ceil

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceil ( numeric_exp )
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling ( numeric_exp )
```

Example 1

```
ceiling ( 4.22 )
```

Result: 5

Example 2

```
ceiling ( -1.23 )
```

Result: -1

char_length

Returns the number of characters in string_exp.

Syntax

```
char_length ( string_exp )
```

Example

```
char_length ( 'Canada' )
```

Result: 6

character_length

Returns the number of characters in string_exp.

Syntax

```
character_length ( string_exp )
```

Example

```
character_length ( 'Canada' )
```

Result: 6

children

Returns the set of children of a specified member.

Syntax

```
children ( member )
```

closingPeriod

Returns the last sibling among the descendants of a member at a specified level. Typically used with a time dimension.

Syntax

```
closingPeriod ( level [, member ] )
```

coalesce

Returns the first non-null argument (or null if all arguments are null). The Coalesce function takes two or more arguments.

Syntax

```
coalesce ( exp_list )
```

completeTuple

Similar to "tuple", identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. However, completeTuple implicitly includes the default member from all dimensions not otherwise specified in the arguments, rather than the current member. The value of this cell can be obtained with the "value" function.

Syntax

```
completeTuple ( member { , member } )
```

cousin

Returns the child member of member2 with the same relative position as the member1 is under its parent.

Syntax

```
cousin ( member1 , member2 )
```

current_date

Returns a date value representing the current date of the computer that the database software runs on.

Syntax

```
current_date
```

Example

```
current_date
```

Result: 2003-03-04

current_time

Returns a time with time zone value, representing the current time of the computer that runs the database software.

Syntax

```
current_time
```

Example

```
current_time
```

Result: 16:33:11+05:00

current_timestamp

Returns a datetime with time zone value, representing the current timestamp of the computer that runs the database software.

Syntax

```
current_timestamp
```

Example

```
current_timestamp
```

Result: 2003-03-03 16:40:15.535000+05:00

currentMember

Returns the current member of the hierarchy during an iteration. If the specified hierarchy is not present in the context in which the expression is being evaluated, its default member is assumed.

Syntax

```
currentMember ( hierarchy )
```

defaultMember

Returns the default member of a hierarchy.

Syntax

```
defaultMember ( hierarchy )
```

descendants

Returns the set of descendants of a member, or set of members, at a specified level (qualified name) or distance (integer 0..n) from the root. Multiple options may be specified (separated by a space) to determine which members are to be returned.

Syntax

```
descendants ( member | set_expr , level | distance [ , { self | before |
beforewithmember | after } ] )
```

Example

```
descendants([national].[Line].[Line].[Line1]->:[PC].[Line (Root)].[Dishwashers]
, 2, SELF AFTER)
```

Result: `Result: AcmeWash MR AcmeWash AcmeWash HE`

Notes

- self: Only the members at the specified level are included in the final set (this is the default behaviour in the absence of any options).
- before: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. If the level specified is the same as the member upon which the function is applied, the member is included in the final set.
- beforewithmember: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. The member upon which the function is applied is also included in the final set.
- after: If other levels exist after the specified level, members from those levels are included in the final set.

emptySet

Returns an empty member set for the specified hierarchy.

Syntax

```
emptySet ( hierarchy )
```

except

Returns the members of "set_exp1" that are not also in "set_exp2". Duplicates are retained only if the optional keyword ALL is supplied as the third argument.

Syntax

```
except ( set_exp1 , set_exp2 [,ALL] )
```

exp

Returns e raised to the power of numeric_exp. The constant e is the base of the natural logarithm.

Syntax

```
exp ( numeric_exp )
```

Example

```
exp ( 2 )
```

Result: 7.389056

extract

Returns an integer representing the value of datepart (year, month, day, hour, minute, second) in datetime_exp.

Syntax

```
extract ( datepart , datetime_exp )
```

Example 1

```
extract ( year , 2003-03-03 16:40:15.535 )
```

Result: 2003

Example 2

```
extract ( hour , 2003-03-03 16:40:15.535 )
```

Result: 16

filter

Returns the set resulting from filtering a specified set based on the boolean condition. Each member is included in the result if and only if the corresponding value of "boolean_exp" is true.

Syntax

```
filter ( set_exp , boolean_exp )
```

firstChild

Returns the first child of a member.

Syntax

```
firstChild ( member)
```

firstSibling

Returns the first child of the parent of a member.

Syntax

```
firstSibling ( member )
```

floor

Returns the largest integer less than or equal to numeric_exp.

Syntax

```
floor ( numeric_exp )
```

Example 1

```
floor ( 3.22 )
```

Result: 3

Example 2

```
floor ( -1.23 )
```

Result: -2

generate

This function evaluates "set_exp2" for each member of "set_exp1" and joins the resulting sets by union. If ALL is specified, duplicates in the result are retained.

Syntax

```
generate ( set_exp1 , set_exp2 [ , ALL ] )
```

head

Returns the first "index_exp" elements of "set_exp". The default for "index_exp" is 1.

Syntax

```
head ( set_exp [ , index_exp ] )
```

hierarchize

This function orders the members of a set in a hierarchy. Members in a level are sorted in their natural order, which is the default ordering of the members along a dimension when no other sort conditions are specified.

Syntax

```
hierarchize ( set_exp )
```

hierarchy

Returns the hierarchy that contains the specified level, member or member set.

Syntax

```
hierarchy ( level | member | set_exp )
```

intersect

Returns the intersection of two input sets. The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Syntax

```
intersect ( set_exp1 , set_exp2 [ , ALL ] )
```

item

Returns a member from a specified location within a set. The index into the set is zero based.

Syntax

```
item ( set_exp , index )
```

lag

Returns the sibling member that is a specified number of positions prior to a specified member.

Syntax

```
lag ( member , index_exp )
```

lastChild

Returns the last child of a specified member.

Syntax

```
lastChild ( member )
```

lastPeriods

Returns the set of members from the same level that ends with the specified member. The number of members returned is the absolute value of "integer_exp". If "integer_exp" is negative, members following and including the specified member are returned. Typically used with a time dimension.

Syntax

```
lastPeriods ( integer_exp , member )
```

lastSibling

Returns the last child of the parent of a specified member.

Syntax

```
lastSibling ( member )
```

lead

Returns the sibling member that is a specified number of positions following a specified member.

Syntax

```
lead ( member , index_exp )
```

level

Returns the level of a member.

Syntax

```
level ( member )
```

levels

Returns the level in the hierarchy whose distance from the root is specified by "index".

Syntax

```
levels ( hierarchy , index )
```

linkMember

Returns the corresponding member in the specified level or hierarchy (of the same dimension). For level-based hierarchies, a level must be specified as the second argument and for parent-child hierarchies, a hierarchy must be specified. An exception is thrown when the second parameter does not resolve to a hierarchy of the dimension that the member of the first parameter belongs to.

Syntax

```
linkMember ( member , level | hierarchy )
```

ln

Returns the natural logarithm of the numeric_exp.

Syntax

```
ln ( numeric_exp )
```

Example

```
ln ( 4 )
```

Result: 1.38629

localtime

Returns a time value, representing the current time of the computer that runs the database software.

Syntax

```
localtime
```

Example

```
localtime
```

Result: 16:33:11

localtimestamp

Returns a datetime value, representing the current timestamp of the computer that runs the database software.

Syntax

```
localtimestamp
```

Example

```
localtimestamp
```

Result: 2003-03-03 16:40:15.535000

lower

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lower ( string_exp )
```

Example

```
lower ( 'ABCDEF' )
```

Result: 'abcdef'

member

Defines a member based on the specified expression in the specified hierarchy. "string1" is used to identify the member created by this function it must be unique in the query, and must be different from any other member in the same hierarchy. "string2" is used as the caption of the member; if it is absent, the caption is empty. If the hierarchy is omitted, the measure dimension is assumed. Note: All calculations used as grouping items whose sibling items are other calculations or member sets, should be explicitly assigned to a hierarchy using this function, otherwise the results are not predictable. The only exception to this is where the calculation involves only members of the same hierarchy as the siblings. In that case the calculation is assumed to belong to that hierarchy.

Syntax

```
member ( value_exp [ , string1 [ , string2 [ , hierarchy ] ] ] )
```

members

Returns the set of members in a hierarchy or level. In the case of a hierarchy, the order of the members in the result is not guaranteed; if a predictable order is required, an explicit ordering function (such as hierarchize) must be used.

Syntax

```
members ( hierarchy | level )
```

mod

Returns the remainder (modulus) of integer_exp1 divided by integer_exp2. The integer_exp2 must not be zero or an exception condition is raised.

Syntax

```
mod ( integer_exp1, integer_exp2 )
```

Example

```
mod ( 20 , 3 )
```

Result: 2

nestedSet

Returns the set of members of set_expr2 evaluated in the context of the current member of set_expr1.

Syntax

```
nestedSet ( set_expr1 , set_expr2 )
```

nextMember

Returns the next member in the level to which the specified member exists.

Syntax

```
nextMember ( member )
```

octet_length

Returns the number of bytes in string_exp.

Syntax

```
octet_length ( string_exp )
```

Example 1

```
octet_length ( 'ABCDEF' )
```

Result: 6

Example 2

```
octet_length ( '' )
```

Result: 0

openingPeriod

Returns the first sibling member among the descendants of a member at a specified level. Typically used with a time dimension.

Syntax

```
openingPeriod ( level [ , member ] )
```

order

Arranges members of a specified set, as determined from the set of values created by evaluating "value_exp" for each value of the set, and modified by the third parameter. There are two varieties of order: hierarchized (ASC or DESC) and non-hierarchized (BASC or BDESC, where B stands for "break hierarchy"). The hierarchized ordering first arranges members according to their position in the hierarchy. Then it orders the children of each member according to "value_exp". The non-hierarchized ordering arranges members in the set without regard to the hierarchy. In the absence of an explicit specification, ASC is the default.

Syntax

```
order ( set_exp , value_exp [ , ASC | DESC | BASC | BDESC ] )
```

ordinal

Returns the zero-based ordinal value (distance from the root level) of the specified level.

Syntax

```
ordinal ( level )
```

parallelPeriod

Returns a member from a prior period in the same relative position as a specified member. This function is similar to the "Cousin" function, but is more closely related to time series. It takes the ancestor of "member" at "level" (call it "ancestor"); then it takes the sibling of "ancestor" that lags by "int_exp" positions, and returns the parallel period of "member" among the descendants of that sibling. When unspecified, "int_exp" defaults to 1 and "member" defaults to the current member.

Syntax

```
parallelPeriod ( level [ , int_exp [ , member ] ] )
```

parent

Returns the member that is the parent of the specified member.

Syntax

```
parent ( member )
```

periodsToDate

Returns a set of sibling members from the same level as a given member, as constrained by a specified level. It locates the ancestor of "member" at "level", and returns that ancestor's descendants at the same level as "member", up to and including "member". Typically used with a time dimension.

Syntax

```
periodsToDate ( level , member )
```

position

Returns the integer value representing the starting position of the first string_exp in the second string_exp or 0 when the first string_exp is not found.

Syntax

```
position ( string_exp , string_exp )
```

Example 1

```
position ( 'C' , 'ABCDEF' )
```

Result: 3

Example 2

```
position ( 'H' , 'ABCDEF' )
```

Result: 0

power

Returns numeric_exp1 raised to the power numeric_exp2. If numeric_exp1 is negative then numeric_exp2 must result in an integer value.

Syntax

```
power ( numeric_exp1, numeric_exp2 )
```

Example

```
power ( 3 , 2 )
```

Result: 9

prevMember

Returns the member that immediately precedes the specified member in the same level.

Syntax

```
prevMember ( member )
```

roleValue

Returns the value of the attribute that is associated with the role whose name is specified by "string" within the specified context. The second argument is optional only in a number of limited circumstances, where it can be derived from other context. Applications can be made portable across different data sources and models by accessing attributes by role, rather than by query item ID. (For dimensionally modelled relational data sources, assignment of roles is the modeller's responsibility.) Intrinsic roles that are defined for members of all data source types include: '_businessKey', '_memberCaption', '_memberDescription', '_memberUniqueName'.

Syntax

```
roleValue ( string [ , member | set_exp ] )
```

Example

```
roleValue ( '_memberCaption', [Sales].[Product].[Product].[Product line]->[all]
.[1] )
```

Result: Camping Equipment

rootMembers

Returns the root members of a hierarchy.

Syntax

```
rootMembers ( hierarchy )
```

set

Returns the list of members defined in the expression. The members must belong to the same hierarchy.

Syntax

```
set ( member { , member } )
```

siblings

Returns the children of the parent of the specified member.

Syntax

```
siblings ( member )
```

sqrt

Returns the square root of numeric_exp. numeric_exp must be non-negative.

Syntax

```
sqrt ( numeric_exp )
```

Example

```
sqrt ( 9 )
```

Result: 3

subset

Returns a subset of members from a specified set starting "index_exp1" from the beginning. If the count "index_exp2" is specified, that many members (if available) are returned. Otherwise, all remaining members are returned.

Syntax

```
subset ( set_exp, index_exp1 [ , index_exp2 ] )
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters or to the end of string_exp if integer_exp2 is omitted. The first character in string_exp is at position 1.

Syntax

```
substring ( string_exp , integer_exp1 [ , integer_exp2 ] )
```

Example

```
substring ( 'abdefg', 3, 2)
```

Result: 'de'

tail

Returns the last "index_exp" elements of "set exp". The default for "index_exp" is 1.

Syntax

```
tail ( set_exp [ , index_exp ] )
```

topCount

This function sorts a set according to the values of "numeric_exp" evaluated at each of the members of "set_exp", and returns the top "index_exp" members.

Syntax

```
topCount ( set_exp , index_exp , numeric_exp )
```

topPercent

This function is similar to topSum, but the threshold is "numeric_exp1" percent of the total.

Syntax

```
topPercent ( set_exp , numeric_exp1, numeric_exp2 )
```

topSum

This function sorts on "numeric_exp2", evaluated at the corresponding members of "set_exp", and picks up the topmost elements whose cumulative total is at least "numeric_exp1".

Syntax

```
topSum ( set_exp , numeric_exp1 , numeric_exp2 )
```

trim

Returns a string_exp trimmed of leading and\or trailing blanks or trimmed of a certain character specified in match_character_exp. BOTH is implicit when first argument is not stated and blank is implicit when second argument is not stated.

Syntax

```
trim ( [ [ TRAILING | LEADING | BOTH ] [ match_character_exp ] , ] string_exp
)
```

Example 1

```
trim ( TRAILING 'A' , 'ABCDEFA' )
```

Result: 'ABCDEF'

Example 2

```
trim ( BOTH ' ABCDEF ' )
```

Result: 'ABCDEF'

tuple

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. Implicitly includes the current member from all dimensions not otherwise specified in the arguments. The current member of any dimension not specified in the evaluating context is assumed to be the default member of that dimension. The value of this cell can be obtained with the "value" function.

Syntax

```
tuple (  member { , member } )
```

union

This function returns the union of 2 sets "set_exp1" and "set_exp2". The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Syntax

```
union ( set_exp1 , set_exp2 [ , ALL ] )
```

unique

Removes all duplicates from the specified set. The remaining members retain their original order.

Syntax

```
unique ( set_expr )
```

upper

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
upper ( string_exp )
```

Example

```
upper ( 'abcdef' )
```

Result: 'ABCDEF'

value

Returns the value of the cell identified by a tuple. Note that the default member of the Measures dimension is the Default Measure.

Syntax

```
value ( tuple )
```

DB2

ascii

Returns the ASCII code value of the leftmost character of the argument as an integer.

Syntax

```
ascii ( string_exp )
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling ( numeric_exp )
```

char

Returns a string representation of a date/time value or a decimal number.

Syntax

```
char ( exp )
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr ( integer_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 with string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

date

Returns a date from a single input value. exp can be a string or integer representation of a date.

Syntax

```
date ( exp )
```

day

Returns the day of the month (1-31) from date_exp. date_exp can be a date value or a string representation of a date.

Syntax

```
day ( date_exp )
```

dayname

Returns a character string containing the data source_specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of date_exp. date_exp can be a date value or a string representation of a date.

Syntax

```
dayname ( date_exp )
```

dayofweek

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Sunday. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofweek ( date_exp )
```

dayofweek_iso

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Monday. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofweek_iso ( date_exp )
```

dayofyear

Returns the day of the year in date_exp as an integer in the range 1 to 366. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofyear ( date_exp )
```

days

Returns an integer representation of a date. exp can be a date value or a string representation of a date.

Syntax

```
days ( exp )
```

dec

Returns decimal representation of string_exp1 with precision numeric_exp1, scale numeric_exp2 and decimal character string_exp2. String_exp1 must be formatted as a SQL Integer or Decimal constant.

Syntax

```
dec ( string_exp1 [ , numeric_exp1 [ , numeric_exp2 [ , string_exp2 ] ] ] )
```

decimal

Returns decimal representation of string_exp1 with precision numeric_exp1, scale numeric_exp2 and decimal character string_exp2. String_exp1 must be formatted as a SQL Integer or Decimal constant.

Syntax

```
decimal ( string_exp1 [ , numeric_exp1 [ , numeric_exp2 [ , string_exp2 ] ] ]
)
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax

```
difference ( string_exp1, string_exp2 )
```

digits

Returns the character string representation of a non-floating point number.

Syntax

```
digits ( numeric_exp )
```

double

Returns the floating-point representation of an expression. 'exp' can be either a numeric or string expression.

Syntax

```
double ( exp )
```

event_mon_state

Returns the operational state of a particular state monitor.

Syntax

```
event_mon_state ( string_exp )
```

float

Returns the floating-point representation of a number.

Syntax

```
float ( numeric_exp )
```

hex

Returns the hexadecimal representation of a value.

Syntax

```
hex ( exp )
```

hour

Returns the hour (an integer from 0, which is midnight, to 23, which is 11:00 pm) from time_exp. time_exp can be a time value or a string representation of a time.

Syntax

```
hour ( time_exp )
```

insert

Returns a string where length (integer_exp2) characters have been deleted from string_exp1 beginning at start (integer_exp1) and where string_exp2 has been inserted into string_exp1 at start. The first character in a string is at position 1.

Syntax

```
insert ( string_exp1, integer_exp1, integer_exp2, string_exp2 )
```

integer

Returns the integer representation of an expression. exp can be a numeric value or a string representation of a number.

Syntax

```
integer ( exp )
```

int

Returns the integer representation of an expression. exp can be a numeric value or a string representation of a number.

Syntax

```
int ( exp )
```

julian_day

Returns an integer value representing the number of days from January 1, 4712 BC (the start of the Julian date calendar) to the date value specified in exp. exp can be a date value or a string representation of a date.

Syntax

```
julian_day ( exp )
```

lcase

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lcase ( string_exp )
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left ( string_exp, integer_exp )
```

length

Returns the length of the operand in bytes (except for double byte string types which return the length in characters).

Syntax

```
length ( exp )
```

locate

Returns the starting position of the first occurrence of string_exp1 within string_exp2. The search starts at position start (integer_exp) of string_exp2. The first character in a string is at position 1. If string_exp1 is not found then zero is returned.

Syntax

```
locate ( string_exp1, string_exp2 [ , integer_exp ] )
```

long_varchar

Returns a long string.

Syntax

```
long_varchar ( string_exp )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim ( string_exp )
```

microsecond

Returns the microsecond (time-unit) part of a value. exp can be a timestamp or a string representation of a timestamp.

Syntax

```
microsecond ( exp )
```

midnight_seconds

Returns an integer value in the range 0 to 86400 representing the number of seconds between midnight and time value specified in the argument. exp can be a time value, a timestamp or a string representation of a time.

Syntax

```
midnight_seconds ( exp )
```

minute

Returns the minute (an integer from 0-59) from time_exp. time_exp can be a time value, a timestamp or a string representation of a time.

Syntax

```
minute ( time_exp )
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month ( date_exp )
```

monthname

Returns a character string containing the data source_specific name of the month (for example, January through December or Jan. through Dec. for a data source that uses English, or Januar through Dezember for a data source that uses German) for the month portion of date_exp.

Syntax

```
monthname ( date_exp )
```

quarter

Returns the quarter in date_exp as a number in the range 1 to 4, where 1 represents January 1 through March 31.

Syntax

```
quarter ( date_exp )
```

radians

Returns the number of radians converted from numeric_exp degrees.

Syntax

```
radians ( numeric_exp )
```

repeat

Returns a string consisting of string_exp repeated integer_exp times.

Syntax

```
repeat ( string_exp, integer_exp )
```

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3.

Syntax

```
replace ( string_exp1, string_exp2, string_exp3 )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right ( string_exp, integer_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round-near (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim ( string_exp )
```

second

Returns the second (an integer from 0-59) from time_exp.

Syntax

```
second ( time_exp )
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

smallint

Returns the small integer representation of a number.

Syntax

```
smallint ( exp )
```

soundex

Returns a 4 character string code obtained by systematically abbreviating words and names in string_exp according to phonetics. Can be used to determine if two strings sound the same, e.g., does sound-of ('SMITH') = sound-of ('SMYTH').

Syntax

```
soundex ( string_exp )
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space ( integer_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

table_name

Returns an unqualified name of a table or view based on the object name in string_exp1 and the schema name given in string_exp2. It is used to resolve aliases.

Syntax

```
table_name ( string_exp1 [ , string_exp2 ] )
```

table_schema

Returns the schema name portion of the two part table or view name based on the object name in string_exp1 and the schema name in string_exp2. It is used to resolve aliases.

Syntax

```
table_schema ( string_exp1 [ , string_exp2 ] )
```

time

Returns a time from a value.

Syntax

```
time ( exp )
```

timestamp

Returns a timestamp from a value or a pair of values. exp1 must represent a date value, and exp2 must represent a time value.

Syntax

```
timestamp ( exp1 [ , exp2 ] )
```

timestamp_iso

Returns a datetime in the ISO format (yyyy-mm-dd hh:mm:ss.nnnnnn) converted from the IBM format (yyyy-mm-dd-hh.mm.ss.nnnnnn). If the exp is a time, it inserts the value of the CURRENT DATE for the date elements and zero for the fractional time element.

Syntax

```
timestamp_iso ( exp )
```

timestampdiff

Returns an estimated number of intervals of type exp1 based on the difference between two timestamps. Exp2 is the result of subtracting two timestamp types and converting the result to CHAR. Valid values of exp1 are: 1 Fractions of a second; 2 Seconds; 4 Minutes; 8 Hours; 16 Days; 32 Weeks; 64 Months; 128 Quarters; 256 Years.

Syntax

```
timestampdiff ( exp1, exp2 )
```

to_char

Returns the string representation of a timestamp with the format of string_exp.

Syntax

```
to_char ( timestamp_exp , string_exp )
```

translate

Returns string_exp1 in which characters from string_exp3 are translated to the equivalent characters in string_exp2. string_exp4 is a single character that is used to pad string_exp2 if it is shorter than string_exp3. If only string_exp1 is present, then this function translates it to uppercase characters.

Syntax

```
translate ( string_exp1 [ , string_exp2, string_exp3 [ , string_exp4 ] ] )
```

trunc

Returns numeric_exp1 truncated to numeric_exp2 places RIGHT of the decimal point. If numeric_exp2 is negative, numeric_exp1 is truncated to the absolute value of numeric_exp2 places to the LEFT of the decimal point.

Syntax

```
trunc ( numeric_exp1, numeric_exp2 )
```

truncate

Returns numeric_exp1 truncated to numeric_exp2 places RIGHT of the decimal point. If numeric_exp2 is negative, numeric_exp1 is truncated to the absolute value of numeric_exp2 places to the LEFT of the decimal point.

Syntax

```
truncate ( numeric_exp1, numeric_exp2 )
```

ucase

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
ucase ( string_exp )
```

value

Returns the first non-null argument (or null if all arguments are null). The Value function takes two or more arguments.

Syntax

```
value ( exp_list )
```

varchar

Returns a VARCHAR representation of exp, with length numeric_exp.

Syntax

```
varchar ( exp [ , numeric_exp ] )
```

week

Returns the week of the year in date_exp as an integer value in the range 1 to 53.

Syntax

```
week ( date_exp )
```

year

Returns the year from date_exp.

Syntax

```
year ( date_exp )
```

DB2 Cast

cast_char

Returns the first numeric_exp characters of the value of exp cast as a string. The whole string is returned when the second argument is not specified.

Syntax

```
cast_char ( exp [ , numeric_exp ] )
```

cast_date

Returns the value of the expression cast as a date.

Syntax

```
cast_date ( exp )
```

cast_decimal

Returns the value of exp cast as a decimal with the precision of numeric_exp1 and scale of numeric_exp2.

Syntax

```
cast_decimal ( exp [ , numeric_exp1, numeric_exp2 ] )
```

cast_double_precision

Returns the value of the expression cast as a double.

Syntax

```
cast_double_precision ( exp )
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float ( exp )
```

cast_integer

Returns the value of the expression cast as a integer.

Syntax

```
cast_integer ( exp )
```

cast_longvarchar

Returns the value of the expression cast as a longvarchar.

Syntax

```
cast_longvarchar ( string_exp )
```

cast_smallint

Returns the value of the expression cast as a smallint.

Syntax

```
cast_smallint ( exp )
```

cast_time

Returns the value of the expression cast as a time value.

Syntax

```
cast_time ( string_exp )
```

cast_timestamp

Returns the value of the expression cast as a datetime.

Syntax

```
cast_timestamp ( exp )
```

cast_varchar

Returns the value of the expression cast as a varchar with length.

Syntax

```
cast_varchar ( exp, integer_exp )
```

DB2 Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log ( numeric_exp )
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand ( integer_exp )
```

DB2 Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
acos ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
asin ( numeric_exp )
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan ( numeric_exp )
```

atanh

Returns the hyperbolic arctangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
atanh (numeric_exp )
```

atan2

Returns the arctangent of the x and y coordinates specified by numeric_exp1 and numeric_exp2, respectively, in radians. The arctangent is the angle whose tangent is numeric_exp2 / numeric_exp1.

Syntax

```
atan2 ( numeric_exp1, numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cosh ( numeric_exp )
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot ( numeric_exp )
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanh ( numeric_exp )
```

Informix

cardinality

Returns the number of elements in a collection column (SET, MULTISET, LIST).

Syntax

```
cardinality ( string_exp )
```

char_length

Returns the number of logical characters (which can be distinct from the number of bytes in some East Asian locales) in string_exp.

Syntax

```
char_length ( string_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

date

Returns the date value of either string_exp or date_exp or int_exp.

Syntax

```
date ( string_exp | date_exp | int_exp )
```

day

Returns an integer that represents the day of the month.

Syntax

```
day ( date_exp )
```

extend

The extend function adjusts the precision of a DATETIME or DATE expression. The expression cannot be a quoted string representation of a DATE value. If you do not specify first and last qualifiers, the default qualifiers are YEAR TO FRACTION(3). If the expression contains fields that are not specified by the qualifiers, the unwanted fields are discarded. If the first qualifier specifies a larger (that is, more significant) field than what exists in the expression, the new fields are filled in with values returned by the CURRENT function. If the last qualifier specifies a smaller field (that is, less significant) than what exists in the expression, the new fields are filled in with constant values. A missing MONTH or DAY field is filled in with 1, and the missing HOUR to FRACTION fields are filled in with 0.

Syntax

```
extend ( date_exp , ' { ' YEAR TO SECOND ' } ' )
```

Example

```
EXTEND (some_date_column , {YEAR TO SECOND} )
```

hex

Returns the hexadecimal encoding of an integer integer_exp.

Syntax

```
hex ( integer_exp )
```

initcap

Returns string_exp, with the first letter of each word in uppercase, all other letters in lowercase. With this function, a word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, colons, and so on, introduce a new word.

Syntax

```
initcap ( string_exp )
```

length

Returns the number of bytes in string_exp, which is not including any trailing blank spaces. For BYTE or TEXT string_exp, LENGTH returns the full number of bytes, including any trailing blank spaces.

Syntax

```
length ( string_exp )
```

lpad

Returns a copy of string_exp1 that is left-padded (string_exp2) to the total number of characters specified by integer_exp. The sequence of string_exp2 occurs as many times as necessary to make the return string the length specified by integer_exp.

Syntax

```
lpad ( string_exp1, integer_exp, string_exp2 )
```

mdy

Returns a type DATE value with three expressions that evaluate to integers that represent the month (integer_exp1), day(integer_exp2), and year(integer_exp3).

Syntax

```
mdy ( integer_exp1, integer_exp2, integer_exp3 )
```

month

Returns an integer corresponding to the month portion of date_exp.

Syntax

```
month ( date_exp )
```

nvl

Returns the value of exp1 if exp1 is not NULL. If exp1 is NULL, then the value of exp2 is returned.

Syntax

```
nvl ( exp1, exp2 )
```

octet_length

Returns the number of bytes in string_exp, including any trailing spaces.

Syntax

```
octet_length ( string_exp )
```

replace

Returns a copy of string_exp1 in which every occurrence of string_exp2 is replaced by string_exp3. If you omit the string_exp3 option, every occurrence of string_exp2 is omitted from the return string.

Syntax

```
replace ( string_exp1, string_exp2 [ , string_exp3 ] )
```

round

Returns the rounded value of an numeric_exp. If you omit the integer_exp, the value is rounded to zero digits or to the units place. The digit range of 32 (+ and -) refers to the entire decimal value.

Syntax

```
round ( numeric_exp [ , integer_exp ] )
```

rpad

Returns a copy of string_exp1 that is right-padded (string_exp2) to the total number of characters specified by integer_exp. The sequence of string_exp2 occurs as many times as necessary to make the return string the length specified by integer_exp.

Syntax

```
rpad ( string_exp1, integer_exp, string_exp2 )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

to_char

Returns the character string of date_exp with the specified string_exp formatting. You can use this function only with built-in data types.

Syntax

```
to_char ( date_exp, string_exp )
```

to_date

Returns the string_exp1 as a date according to the date format you specify in the string_exp2 parameter. If string_exp1 is NULL, then a NULL value is returned.

Syntax

```
to_date ( string_exp1, string_exp2 )
```

trunc

Returns the truncated value of a numeric_exp. If you omit integer_exp, the value is truncated to zero digits or to the unit's place. The digit limitation of 32 (+ and -) refers to the entire decimal value.

Syntax

```
trunc ( numeric_exp [ , integer_exp ] )
```

weekday

Returns an integer that represents the day of the week; zero (0) represents Sunday, one (1) represents Monday, and so on.

Syntax

```
weekday ( date_exp )
```

year

Returns a four-digit integer that represents the year.

Syntax

```
year ( date_exp )
```

Informix Math

log10

Returns the log of a numeric_exp to base 10.

Syntax

```
log10 ( numeric_exp )
```

logn

Returns the natural logarithm of a numeric_exp.

Syntax

```
logn ( numeric_exp )
```

root

Returns the root value of a numeric_exp. Requires at least one numeric argument (the radians argument). If only the numeric_exp1 is supplied, the value 2 is used as a default value for numeric_exp2; 0 cannot be used as the value of numeric_exp2.

Syntax

```
root ( numeric_exp1[ , numeric_exp2 ] )
```

Informix Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
acos ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
asin ( numeric_exp )
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan ( numeric_exp )
```

atan2

Returns the arctangent of the x and y coordinates specified by numeric_exp1 and numeric_exp2, respectively, in radians. The arctangent is the angle whose tangent is numeric_exp1.

Syntax

```
atan2 ( numeric_exp1, numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

MS Access

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp.

Syntax

```
ascii(string_exp)
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling(numeric_exp)
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr(integer_exp)
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat(string_exp1, string_exp2)
```

curdate

Returns a date value representing the current date of the computer that the database software runs on.

Syntax

```
curdate()
```

curtime

Returns a time value representing the current time of the computer that the database software runs on.

Syntax

```
curtime()
```

dayname

Returns a character string containing the data source_specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of date_exp.

Syntax

```
dayname(date_exp)
```

dayofmonth

Returns the day of the month (1-31) from date_exp. Returns the days field (a signed integer) from interval_exp.

Syntax

```
dayofmonth(date_exp|interval_exp)
```

dayofweek

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Monday.

Syntax

```
dayofweek(date_exp)
```

dayofyear

Returns the day of the year in date_exp as an integer in the range 1 to 366.

Syntax

```
dayofyear(date_exp)
```

hour

Returns the hour (an integer from 0, which is midnight, to 23, which is 11:00 pm) from time_exp.

Syntax

```
hour(time_exp)
```

instr

Searches string_exp1 for the first occurrence of string_exp2. Returns an integer specifying the position of string_exp2. The optional argument integer_exp1 sets the starting position for the search. If omitted, the search begins at the first character position of string_exp1. The optional argument integer_exp2 specifies the type of string comparison. integer_exp1 is required if integer_exp2 is specified.

Syntax

```
instr ( [ integer_exp1 , ] string_exp1, string_exp2 [ , integer_exp2 ] )
```

lcase

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lcase(string_exp)
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left(string_exp, integer_exp)
```

length

Returns the number of characters in string_exp, excluding trailing blanks and the string termination character.

Syntax

```
length(string_exp)
```

locate

Returns the starting position of the first occurrence of string_exp1 within string_exp2. The search starts at position start (integer_exp) of string_exp2. The first character in a string is at position 1. If string_exp1 is not found then zero is returned.

Syntax

```
locate(string_exp1, string_exp2 [ , integer_exp ] )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim(string_exp)
```

minute

Returns the minute (an integer from 0-59) from time_exp.

Syntax

```
minute(time_exp)
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month(date_exp)
```

monthname

Returns a character string containing the data source_specific name of the month (for example, January through December or Jan. through Dec. for a data source that uses English, or Januar through Dezember for a data source that uses German) for the month portion of date_exp.

Syntax

```
monthname(date_exp)
```

now

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax

```
now()
```

position

Returns the starting position of string_exp1 in string_exp2. The first character in a string is at position 1.

Syntax

```
position(string_exp1, string_exp2)
```

quarter

Returns the quarter in date_exp as a number in the range 1 to 4, where 1 represents January 1 through March 31.

Syntax

```
quarter(date_exp)
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right(string_exp, integer_exp)
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point.

Syntax

```
round(numeric_exp, integer_exp)
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim(string_exp)
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign(numeric_exp)
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space(integer_exp)
```

substr

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substr(string_exp, integer_exp1, integer_exp2)
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substring(string_exp, integer_exp1, integer_exp2)
```

truncate

Returns string_exp with trailing spaces removed.

Syntax

```
truncate(string_exp)
```

ucase

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
ucase(string_exp)
```

week

Returns the week of the year in date_exp as an integer value in the range 1 to 53.

Syntax

```
week(date_exp)
```

year

Returns the year from date_exp.

Syntax

```
year(date_exp)
```

MS Access Cast

cast_decimal

Returns the value of the expression cast as a decimal.

Syntax

```
cast_decimal(exp)
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float(exp)
```

cast_integer

Returns the value of the expression cast as a integer.

Syntax

```
cast_integer(exp)
```

cast_numeric

Returns the value of string_exp cast as a numeric value.

Syntax

```
cast_numeric(string_exp)
```

cast_real

Returns the value of the expression cast as a real.

Syntax

```
cast_real(exp)
```

cast_smallint

Returns the value of the expression cast as a smallint.

Syntax

```
cast_smallint(exp)
```

cast_varchar

Returns the value of the expression cast as a varchar.

Syntax

```
cast_varchar(exp)
```

MS Access Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log(numeric_exp)
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand(integer_exp)
```

MS Access Trigonometry

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan(numeric_exp)
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos(numeric_exp)
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin(numeric_exp)
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan(numeric_exp)
```

Oracle

add_months

Returns the datetime resulting from adding integer_exp months to date_exp.

Syntax

```
add_months ( date_exp, integer_exp )
```

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp, e.g. ascii ('A') is 65.

Syntax

```
ascii ( string_exp )
```

ceil

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceil ( numeric_exp )
```

char_length

Returns the number of characters in string_exp.

Syntax

```
char_length ( string_exp )
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr ( integer_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

decode

DECODE compares expr to each search value one by one. If expr is equal to a search, then returns the corresponding result. If no match is found, then returns default. If default is omitted, then returns null.

Syntax

```
decode ( expr , search , result [, search , result]... [, default] )
```

dump

Returns internal representation of 'expr' with the format of numeric_exp1 starting from position numeric_exp2 for numeric_exp3.

Syntax

```
dump ( expr [ , numeric_exp1 [ , numeric_exp2 [ , numeric_exp3 ] ] ] )
```

greatest

Returns the greatest value in a list of expressions.

Syntax

```
greatest ( exp_list )
```

initcap

Returns string_exp, with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

Syntax

```
initcap ( string_exp )
```

instr

Searches string_exp1 from the integer_exp1 position for the (integer_exp2)th occurance of string_exp2. If integer_exp1 is negative then the search is backwards from the end of string_exp1. Returns an integer indicating the position of string_exp2.

Syntax

```
instr ( string_exp1, string_exp2 [ , integer_exp1 [ , integer_exp2 ] ] )
```

instrb

Searches string_exp1 from the integer_exp1 position for the (integer_exp2)th occurance of string_exp2. If integer_exp1 is negative then the search is backwards from the end of string_exp1. The result returned indicates the position (byte number) where search was found.

Syntax

```
instrb ( string_exp1, string_exp2 [ , integer_exp1 [ , integer_exp2 ] ] )
```

least

Returns the least value in a list of expressions.

Syntax

```
least ( exp_list )
```

length

Returns the number of characters in string_exp.

Syntax

```
length ( string_exp )
```

lengthb

Returns the number of bytes in string_exp.

Syntax

```
lengthb ( string_exp )
```

lpad

Returns string_exp1 padded to length integer_exp with occurrences of string_exp2. If string_exp1 is longer than integer_exp then returns the appropriate portion of string_exp1.

Syntax

```
lpad ( string_exp1, integer_exp [ , string_exp2 ] )
```

ltrim

Returns string_exp1, with leading characters removed up to the first character not in string_exp2, e.g. ltrim('xyxXxyAB', 'xy') returns 'XxyAB'.

Syntax

```
ltrim ( string_exp1 [ , string_exp2 ] )
```

months_between

Returns the number of months from date_exp1 to date_exp2. If date_exp1 is later than date_exp2 then the result will be a positive number. The days and time portion of the difference are ignored, i.e. the months are not rounded, except if date_exp1 and date_exp2 are the last days of a month.

Syntax

```
months_between ( date_exp1, date_exp2 )
```

new_time

Returns the Datetime in timezone 'new_tz' for 'datetime' in 'old_tz' timezone. 'Old_tz' and 'new_tz' can be one of 'AST', 'ADT', 'BST', 'BDT', 'CST', 'CDT', 'EST', 'EDT', 'HST', 'HDT', 'MST', 'MDT', 'NST', 'PST', 'PDT', 'YST' or 'YDT'.

Syntax

```
new_time ( datetime_exp, old_tz, new_tz )
```

next_day

Returns the datetime of the first weekday named by string_exp that is later than datetime_exp. The return value has the same hours, minutes, and seconds as datetime_exp.

Syntax

```
next_day ( datetime_exp, string_exp )
```

nls_initcap

Returns string_exp1 with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric. string_exp2 specifies the sorting sequence.

Syntax

```
nls_initcap ( string_exp1 [ , string_exp2 ] )
```

nls_lower

Returns string_exp1 with all letters in lowercase. string_exp2 specifies the sorting sequence.

Syntax

```
nls_lower ( string_exp1 [ , string_exp2 ] )
```

nls_upper

Returns string_exp1 with all letters in uppercase. string_exp2 specifies the sorting sequence.

Syntax

```
nls_upper ( string_exp1 [ , string_exp2 ] )
```

nvl

If exp is null (missing) returns constant. If exp is not null returns exp. Valid for numeric_exp, string_exp, date_exp, and time_exp.

Syntax

```
nvl ( exp, constant )
```

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3. If string_exp3 is not specified then it replaces all occurrences with null (ie: removes all occurances of string_exp2).

Syntax

```
replace ( string_exp1, string_exp2 [ , string_exp3 ] )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp [ , integer_exp ] )
```

rpad

Returns string_exp1 right-padded to length integer_exp with occurrences of string_exp2. If string_exp1 is longer than integer_exp then returns the appropriate portion of string_exp1. If string_exp2 is not specified then spaces are used.

Syntax

```
rpad ( string_exp1, integer_exp [ , string_exp2 ] )
```

rtrim

Returns string_exp1, with final characters removed after the last character not in string_exp2, e.g. rtrim('ABxXxyx', 'xy') returns 'ABxX'. If string_exp2 is not specified it removes th final space characters.

Syntax

```
rtrim ( string_exp1 [ , string_exp2 ] )
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

soundex

Returns a character string containing the phonetic representation of string_exp.

Syntax

```
soundex ( string_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

substrb

Same as substr, except that the arguments are expressed in bytes (not characters).

Syntax

```
substrb ( string_exp, numeric_exp1 [ , numeric_exp2 ] )
```

{sysdate}

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax

```
{ sysdate }
```

to_char

Returns the string representation of exp with the format of string_exp. exp can either be a date value or a numeric value.

Syntax

```
to_char ( exp [ , string_exp ] )
```

to_date

Converts string_exp1 to a datetime value as specified by the format string_exp2. string_exp3 specifies format elements such as language.

Syntax

```
to_date ( string_exp1 [ , string_exp2 [ , string_exp3 ] ] )
```

to_number

Converts string_exp1 to a numeric value as specified by the format string_exp2. string_exp3 specifies format elements such as currency information.

Syntax

```
to_number ( string_exp1, string_exp2, string_exp3 )
```

translate

Returns string_exp1, with all occurrences of each character in string_exp2 replaced by its corresponding character in string_exp3.

Syntax

```
translate ( string_exp1, string_exp2, string_exp3 )
```

trunc

Truncates the date_exp using the format specified by string_exp. For example, if string_exp is 'YEAR' then date_exp is truncated to the first day of the year.

Syntax

```
trunc ( date_exp, string_exp )
```

trunc

Truncates digits from numeric_exp1 using numeric_exp2 as the precision.

Syntax

```
trunc ( numeric_exp1, numeric_exp2 )
```

{user}

Returns the username of the current Oracle user.

Syntax

```
{ user }
```

vsize

Returns the number of bytes in the internal representation of 'exp'. 'exp' must be a string expression.

Syntax

```
vsize ( exp )
```

Oracle Math

log

Returns the logarithm of numeric_exp2 to the base numeric_exp1.

Syntax

```
log ( numeric_exp1, numeric_exp2 )
```

Oracle Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
acos ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
asin ( numeric_exp )
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan ( numeric_exp )
```

atan2

Returns the arctangent of the x and y coordinates specified by numeric_exp1 and numeric_exp2, respectively, in radians. The arctangent is the angle whose tangent is numeric_exp2 / numeric_exp1.

Syntax

```
atan2 ( numeric_exp1 ,numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cosh ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanh ( numeric_exp )
```

Red Brick

ceil

Returns the smallest integer greater than or equal to numeric_exp or string_exp. Note that string_exp must represent a valid numeric value.

Syntax

```
ceil ( numeric_exp | string_exp )
```

concat

This function concatenates character strings and returns the concatenated string of characters.

Syntax

```
concat ( string_exp1 , string_exp2 )
```

{current_user}

Returns the database username (authorization ID) of the current user.

Syntax

```
{ current_user }
```

date

This function creates a date value from a character string or a timestamp expression and returns a date data type. The expression can be either characters or timestamp.

Syntax

```
date ( expression )
```

dateadd

This function adds an interval to a datetime value and returns a result that is the same datetime data type as that of datetime_expression. The datepart refers to year, month, day, hour, minute, second. The interval must be an integer and datetime_exp can be date, time or timestamp.

Syntax

```
dateadd ( { datepart }, interval, datetime_exp )
```

datediff

This function finds the difference between two datetime expressions and returns an integer result in datepart units. The datepart refers to year, month, day, hour, minute, second. The datetime_exp can be date, time or timestamp.

Syntax

```
datediff ( { datepart }, datetime_exp, datetime_exp )
```

datename

This function extracts the specified datepart component and returns its value as a character string. The datepart refers to year, month, day, hour, minute, second. The datetime_exp can be date, time or timestamp.

Syntax

```
datename ( { datepart }, datetime_exp )
```

dec

This function converts a specified value to a decimal value and returns a value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

Syntax

```
dec ( expression, [precision, scale] )
```

decimal

This function converts a specified value to a decimal value and returns a value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

Syntax

```
decimal ( expression, [precision, scale] )
```

decode

This function compares and converts an expression to another value. If the expression matches target, it is replaced by the corresponding replacement; otherwise the expression is replaced by default or by NULL if no default is specified. The expressions can be any data type and all expressions must be the same data type.

Syntax

```
decode ( expression, target, replacement [,default] )
```

float

This function converts a specified value into a double-precision floating-point value.

Syntax

```
float ( numeric_exp )
```

ifnull

This function tests an expression for missing values and replaces each one with a specified value. If expression is NULL, this function returns substitute; otherwise it returns the value of the expression. The expressions can be any data type and all expressions must be the same data type.

Syntax

```
ifnull ( expression , substitute )
```

int

This function converts a specified numeric string into an integer value and returns an integer value. If the argument is null, this function returns NULL.

Syntax

```
int ( numeric_exp )
```

integer

This function converts a specified numeric string into an integer value and returns an integer value. If the argument is null, this function returns NULL.

Syntax

```
integer ( numeric_exp )
```

length

If the argument is not null, this function returns an integer result specifying the number of characters in the string; otherwise the result is NULL.

Syntax

```
length ( string_exp )
```

lengthb

If the argument is not null, this function returns an integer result specifying the number of bytes in the string. If the argument is null, the result is NULL.

Syntax

```
lengthb ( string_exp )
```

ltrim

If the argument is not null, this function removes leading blanks from the character string; otherwise the result is NULL.

Syntax

```
ltrim ( string_exp )
```

nullif

This function returns NULL if both expressions have the same value. If the expressions have different values, the value of the first expression is returned. The exp1 and exp2 can be any data type and must be the same data type.

Syntax

```
nullif ( exp1, exp2 )
```

positionb

If the first string_exp is located, this function returns an integer that is relative to the beginning byte position of the first string_exp in the second string_exp. If the first string_exp is not located, the result is 0. If the first string_exp is of zero length, the result is 1. If the first string_exp is null, an error message is returned. If the second string_exp is null, the result is 0.

Syntax

```
positionb ( string-exp, string_exp )
```

real

This function returns a real value. If the argument is null, this function returns NULL.

Syntax

```
real ( numeric_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

If the argument is not null, this function removes trailing blanks from the character string; otherwise the result is NULL.

Syntax

```
rtrim ( string_exp )
```

sign

This function calculates the sign of the expression, and returns 1 for a positive value, –1 for a negative value, and 0 for zero.

Syntax

```
sign ( numeric_exp )
```

string

This function converts numeric or datetime values to character strings. The expression can be numeric or datetime.

Syntax

```
string ( expression [, length [, scale]] )
```

substr

If the first argument is not null, this function returns the substring that begins at position start and continues for length characters. If length is not specified, this function returns a substring from start to the end of string_exp.

Syntax

```
substr ( string_exp, start_integer, length_integer )
```

substrb

If the first argument is not null, this function returns the substring that begins at position start and continues for length bytes. If length is not specified, this function returns a substring from start to the end of string_exp.

Syntax

```
substrb ( string_exp, start_integer, length_integer )
```

time

This function creates a time value from a character string or a time-stamp data type expression.

Syntax

```
time ( expression )
```

timestamp

This function creates a time-stamp value from a character string.

Syntax

```
timestamp ( timestamp_exp )
```

timestamp

This function creates a time-stamp value from time and date values. If there are two arguments, the first must be a date expression and the second must be a time expression, separated by a comma (,). If either the date expression or the time expression is null, the resulting time-stamp expression is also null.

Syntax

```
timestamp ( date_exp, time_exp )
```

to_char

This function is a datetime scalar function that operates on a date, time, or timestamp data type and returns the character string specified by a given format.

Syntax

```
to_char ( source_date, format_str )
```

SQL Server

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp, e.g. ascii ('A') is 65.

Syntax

```
ascii(string_exp)
```

char

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255. For example, char(65) has the value 'A'.

Syntax

```
char(integer_exp)
```

charindex

Searches string_exp2 for the first occurrence of string_exp1 and returns an integer. start_location is the character position to start searching for string_exp1 in string_exp2. If start_location is not given, is a negative number, or is zero, the search starts at the beginning of string_exp2.

Syntax

```
charindex ( string_exp1, string_exp2 [ , start_location ]  )
```

{current_user}

Returns the name of the current user.

Syntax

```
{ current_user }
```

datalength

Returns the length in bytes of the string.

Syntax

```
datalength(string_exp)
```

dateadd

Returns the date resulting from adding integer_exp units indicated by datepart(e.g. day, month, year) to date_exp.

Syntax

```
dateadd({datepart}, integer_exp, date_exp)
```

datediff

Returns the number of units indicated by datepart(e.g. day, month, year) between date_exp1 and date_exp2.

Syntax

```
datediff({datepart}, date_exp1, date_exp2)
```

datename

Returns part of a datetime, smalldatetime, date or time value as an ASCII string. Note that the datepart argument must be a keyword representing a datepart or its abbreviation recognized by Microsoft SQL Server and must be enclosed in curly brackets.

Syntax

```
datename ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datename ( {mm}, 2000-01-01 )
```

Result: January.

datepart

Returns part of a datetime, smalldatetime, date or time value (for example, the month) as an integer. Note that the datepart argument must be a keyword representing a datepart or its abbreviation recognized by Microsoft SQL Server and must be enclosed in curly brackets.

Syntax

```
datepart ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datepart ( {wk}, 2000-01-01 )
```

Result: 1 (first week of the year).

day

Returns the day portion of date_exp. Same as extract(day from date_exp).

Syntax

```
day(date_exp)
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax

```
difference(string_exp1, string_exp2)
```

getdate

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax

```
getdate()
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left(string_exp, integer_exp)
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim(string_exp)
```

month

Returns the month portion of date_exp. Same as extract(month from date_exp).

Syntax

```
month(date_exp)
```

patindex

Returns an integer which represents the starting position if the first occurrence of string_exp1 is in the string_exp2; returns 0 if string-exp1 is not found. The % wildcard character must precede and follow pattern in string_exp1, except when searching for first or last characters.

Syntax

```
patindex ( string_exp1, string_exp2 )
```

Example

```
patindex ( '%nos%', 'Cognos' )
```

Result: 4

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3.

Syntax

```
replace ( string_exp1 , string_exp2 , string_exp3 )
```

replicate

Returns a string consisting of string_exp repeated integer_exp times.

Syntax

```
replicate(string_exp, integer_exp)
```

reverse

Returns the reverse of the character expression.

Syntax

```
reverse ( string_exp )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right(string_exp, integer_exp)
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point.

Syntax

```
round(numeric_exp,integer_exp)
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim(string_exp)
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign(numeric_exp)
```

soundex

Returns a four character string representing the sound of the words in string_exp.

Syntax

```
soundex(string_exp)
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space(integer_exp)
```

str

Returns a string representation of numeric_exp. integer_exp1 is the length of the string returned. integer_exp2 is the number of decimal digits.

Syntax

```
str(numeric_exp [ , integer_exp1 [ , integer_exp2 ] ] )
```

stuff

Returns a string where length (integer_exp2) characters have been deleted from string_exp1 beginning at start (integer_exp1) and where string_exp2 has been inserted into string_exp1 at start. The first character in a string is at position 1.

Syntax

```
stuff(string_exp1, integer_exp1, integer_exp2, string_exp2)
```

year

Returns the year portion of date_exp. Same as extract(year from date_exp).

Syntax

```
year(date_exp)
```

SQL Server Cast

cast_char

Returns the value of the expression cast as a char. A limit of 30 characters is returned.

Syntax

```
cast_char(exp)
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float(exp)
```

cast_integer

Returns the value of the expression cast as an integer.

Syntax

```
cast_integer(exp)
```

cast_real

Returns the value of the expression cast as a real.

Syntax

```
cast_real(exp)
```

cast_smallint

Returns the value of the expression cast as a small integer.

Syntax

```
cast_smallint(exp)
```

SQL Server Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log(numeric_exp)
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10(numeric_exp)
```

pi

Returns the constant value of pi as a floating point value.

Syntax

```
pi()
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand(integer_exp)
```

SQL Server Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
acos(numeric_exp)
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
asin(numeric_exp)
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan(numeric_exp)
```

atn2

Returns the arctangent of the x and y coordinates specified by numeric_exp1 and numeric_exp2, respectively, in radians. The arctangent is the angle whose tangent is numeric_exp1.

Syntax

```
atn2(numeric_exp1, numeric_exp2)
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos(numeric_exp)
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot(numeric_exp)
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees(numeric_exp)
```

radians

Returns the number of radians converted from numeric_exp degrees.

Syntax

```
radians(numeric_exp)
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin(numeric_exp)
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan(numeric_exp)
```

Teradata

account

This function returns the account string for the current user.

Syntax

```
{account}
```

add_months

Returns the date or the datetime resulting from adding integer_exp months to date_exp or datetime_exp.

Syntax

```
add_months ( date_exp | datetime_exp, integer_exp )
```

bytes

This function returns the number of bytes contained in the specified byte string.The byte_exp are restricted to BYTE or VARBYTE.

Syntax

```
bytes ( byte_exp )
```

case_n

This function evaluates a list of conditions and returns the position of the first condition that evaluates to TRUE, provided that no prior condition in the list evaluates to UNKNOWN. The keywords must be enclosed in curly brackets. The NO CASE is an optional condition that evaluates to TRUE if every conditional_expression in the list evaluates to FALSE. The NO CASE OR UNKNOWN condition evaluates to TRUE if every conditional_expression in the list evaluates to FALSE, or if a conditional_expression evaluates to UNKNOWN and all prior conditions in the list evaluate to FALSE. The UNKNOWN is an optional condition that evaluates to TRUE if a conditional_expression evaluates to UNKNOWN and all prior conditions in the list evaluate to FALSE.

Syntax

```
case_n ( condition_exp_list [, NO CASE | UNKNOWN | NO CASE OR UNKNOWN [, UNKNOWN
 ]] )
```

char2hexint

This function returns the hexadecimal representation for a character string.

Syntax

```
char2hexint ( string_exp )
```

characters

This function returns an integer value representing the number of logical characters or bytes contained in the specified operand string.

Syntax

```
characters ( string_exp )
```

database

This function returns the name of the default database for the current user.

Syntax

```
{database}
```

date

This function returns the current date.

Syntax

```
{date}
```

format

This function returns the declared format for the named expression. The data type returned by a FORMAT phrase is a variable character string of up to 30 characters.

Syntax

```
format ( expression )
```

index

This function returns the position in string_exp1 where string_exp2 starts.

Syntax

```
index ( string_exp1, string_exp2 )
```

log

Computes the base 10 logarithm of an argument. The numeric_exp is a non-zero, positive numeric expression.

Syntax

```
log ( numeric_exp )
```

nullif

This function returns NULL if scalar_exp1 and scalar_exp2 are equal. Otherwise, it returns its first argument, scalar_exp1. The scalar_exp1 and scalar_exp2 can be any data type.

Syntax

```
nullif ( scalar_exp1, scalar_exp2 )
```

nullifzero

This function converts data from zero to null to avoid problems with division by zero.

Syntax

```
nullifzero ( numeric_exp )
```

profile

This function returns the current profile for the session or NULL if none.

Syntax

```
{profile}
```

random

This function returns a random integer number for each row of the results table. The lower_bound and upper_bound are integer constants. The limits for lower_bound, upper_bound range from -2147483648 to 2147483647, inclusive. The upper_bound must be greater than or equal to lower_bound.

Syntax

```
random ( lower_bound, upper_bound )
```

role

This function returns the current role for the session or NULL if none.

Syntax

```
{role}
```

session

This function returns the number of the session for the current user.

Syntax

```
{session}
```

soundex

This function returns a character string that represents the Soundex code for string_exp.

Syntax

```
soundex ( string_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

time

This function returns the current time based on a 24-hour day.

Syntax

```
{time}
```

type

This function returns the data type defined for an expression.

Syntax

```
type ( expression )
```

user

This function returns the user name of the current user.

Syntax

```
{user}
```

vargraphic

This function returns a character string that represents the vargraphic code for string_exp.

Syntax

```
vargraphic ( string_exp )
```

zeroifnull

This function converts data from null to 0 to avoid cases where a null result creates an error. If the numeric_exp is not null, it returns the value of the numeric_exp, if numeric_exp is a character string, it is converted to a numeric value of FLOAT data type. If the numeric_exp is null or zero, it returns zero.

Syntax

```
zeroifnull ( numeric_exp )
```

Teradata Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp. The values of numeric_exp must be between -1 and 1, inclusive.

Syntax

```
acos ( numeric_exp )
```

acosh

Returns the inverse hyperbolic cosine of an argument. The numeric_exp can be any real number equal to or greater than 1.

Syntax

```
acosh ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp. The values of numeric_exp must be between -1 and 1, inclusive.

Syntax

```
asin ( numeric_exp )
```

asinh

Returns the inverse hyperbolic sine of an argument. The numeric_exp can be any real number.

Syntax

```
asinh ( numeric_exp )
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan ( numeric_exp )
```

atan2

Returns the arctangent of the x and y coordinates specified by numeric_exp1 and numeric_exp2, respectively, in radians. ATAN2(x,y) equals ATAN(y/x), except that x can be 0 in ATAN2(x,y). The returned angle is between - and π radians, excluding π.

Syntax

```
atan2 ( numeric_exp1, numeric_exp2 )
```

atanh

Returns the inverse hyperbolic tangent of an argument. The numeric_exp can be any real number between 1 and -1, excluding 1 and -1.

Syntax

```
atanh (numeric_exp )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of an argument. The numeric_exp can be any real number.

Syntax

```
cosh ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of an argument. The numeric_exp can be any real number.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of an argument. The numeric_exp can be any real number.

Syntax

```
tanh ( numeric_exp )
```

SAP BW

SAP BW Trigonometry

arccos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
arccos ( numeric_exp )
```

arcsin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
arcsin ( numeric_exp )
```

arctan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
arctan ( numeric_exp )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

coshyp

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
coshyp ( numeric_exp )
```

sinhyp

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinhyp ( numeric_exp )
```

tanhyp

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanhyp ( numeric_exp )
```

SAP BW Math

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

Sybase

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp.

Syntax

```
ascii ( string_exp )
```

Example

```
ascii( 'A' )
```

Result: 65

char

Converts a single-byte integer value to a character value. char is usually used as the inverse of ascii. integer_exp must be between 0 and 255. Returns a char datatype. If the resulting value is the first byte of a multibyte character, the character may be undefined.

Syntax

```
char ( integer_exp )
```

charindex

Searches string_exp2 for the first occurrence of string_exp1and returns an integer, which represents its starting position. If string_exp1 is not found, it returns 0. If string_exp1 contains wildcard characters, charindex treats it them as literals.

Syntax

```
charindex ( string_exp1, string_exp2 )
```

datalength

Returns the length in bytes of the string.

Syntax

```
datalength ( string_exp )
```

dateadd

Returns the date resulting from adding integer_exp units indicated by datepart(e.g. day, month, year) to date_exp. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
dateadd ( ' { ' datepart ' } ' , integer_exp, date_exp )
```

Example

```
dateadd ( {dd}, 16, 1997-06-16 )
```

Result: Jul 2, 1997

datediff

Returns the number of units indicated by datepart(e.g. day, month, year) between date_exp1 and date_exp2. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datediff ( ' { ' datepart ' } ' , date_exp1, date_exp2 )
```

Example

```
datediff ( {yy}, 1984-01-01, 1997-01-01 )
```

Result: 13

datename

Returns part of a datetime, smalldatetime, date or time value as an ASCII string. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datename ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datename ( {mm}, 1999-05-01 )
```

Result: May

datepart

Returns part of a datetime, smalldatetime, date or time value (for example, the month) as an integer. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datepart ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datepart ( {mm}, 1999-05-01 )
```

Result: 5

day

Returns the day of the month (1-31) from date_exp.

Syntax

```
day ( date_exp )
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax

```
difference ( string_exp1, string_exp2 )
```

getdate

Returns current system date and time.

Syntax

```
getdate ()
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left ( string_exp, integer_exp )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim ( string_exp )
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month ( date_exp )
```

patindex

Returns an integer which represents the starting position if the first occurrence of string_exp1 is in the string_exp2; returns 0 if string-exp1 is not found. By default, patindex returns the offset in characters. To return offset in bytes, that is, multibyte character string, specify using bytes. The % wildcard character must precede and follow pattern in string_exp1, except when searching for first or last characters

Syntax

```
patindex ( string_exp1, string_exp2 [ using {bytes | chars | characters} ] )
```

rand

Returns a random float value between 0 and 1, using the optional integer as a seed value.

Syntax

```
rand ( integer_exp )
```

replicate

Returns a string with the same datatype as string_exp, containing the same expression repeated integer_exp times or as many times as will fit into a 225-byte space, whichever is less.

Syntax

```
replicate ( string_exp, integer_exp )
```

reverse

Returns the reverse of the character or binary expression; if string_exp is "abcd", it returns "dcba".

Syntax

```
reverse ( string_exp )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right ( string_exp, integer_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim ( string_exp )
```

soundex

Returns a four-character soundex code for character strings that are composed of a contigous sequence of valid single- or double byte Roman letter.

Syntax

```
soundex ( string_exp )
```

space

Returns a string with the indicated number of single-byte space.

Syntax

```
space ( integer_exp )
```

str

Returns a string representation of numeric_exp. integer_exp1 is the length of the string returned. integer_exp2 is the number of decimal digits. length and decimal are optional(default length is 10; default decimal is 0.)

Syntax

```
str ( numeric_exp [ , integer_exp1 [ , integer_exp2 ] ] )
```

stuff

Delete integer_exp2 characters from string_exp1 at integer_exp1, and then insert string_exp2 into string_exp1 at integer_exp. To delete characters without inserting other characters, string_exp2 should be NULL, not " ", which indicates a single space.

Syntax

```
stuff ( string_exp1, integer_exp1, integer_exp2, string_exp2 )
```

substring

Returns the substring of string_exp that starts at position integer_exp1. integer_exp2 specifies the number of characters in the substring.

Syntax

```
substring ( string_exp, integer_exp1, integer_exp2 )
```

to_unichar

Returns a unichar expression having the value of the integer_exp. If the integer_exp is in the range 0xD800..0xDFFF, the operation is aborted. If the integer_exp is in the range 0..0xFFFF, a single Unicode value is returned. If the integer_exp is in the range 0x10000..0x10FFFF, a surrogate pair is returned.

Syntax

```
to_unichar ( integer_exp )
```

uhighsurr

Returns 1 if the Unicode value at integer_exp is the high half of a surrogate pair (which should appear first in the pair). Otherwise, returns 0. This function allows you to write explicit code for surrogate handling. Particularly, if a substring starts on a Unicode character where uhighsurr() is true, extract a substring of at least 2 Unicode values, as substr() does not extract just 1. substr() does not extract half of a surrogate pair.

Syntax

```
uhighsurr ( string_exp, integer_exp )
```

ulowsurr

Returns 1 if the Unicode value at integer_exp is the low half of a surrogate pair (which should appear second in the pair). Otherwise, returns 0. This function allows you to explicitly code around the adjustments performed by substr(), stuff(), and right(). Particularly, if a substring ends on a Unicode value where ulowsurr() is true, extract a substring of 1 less characters (or 1 more), since substr() does not extract a string that contains an unmatched surrogate pair.

Syntax

```
ulowsurr ( string_exp, integer_exp )
```

uscalar

Returns the Unicode scalar value for the first Unicode character in string_exp. If the first character is not the high-order half of a surrogate pair, then the value is in the range 0..0xFFFF. If the first character is the high-order half of a surrogate pair, a second value must be a low-order half, and the return value is in the range 0x10000..0x10FFFF. If this function is called on a uchar_expr containing an unmatched surrogate half and the operation aborted.

Syntax

```
uscalar ( string_exp )
```

year

Returns the year from date_exp.

Syntax

```
year ( date_exp )
```

Sybase Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log ( numeric_exp )
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

pi

Returns the constant value of pi as a floating point value.

Syntax

```
pi ()
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

Sybase Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
acos ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
asin ( numeric_exp )
```

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

atn2

Returns the angle (in radians) whose tangent is (numeric_exp1/numeric_exp2).

Syntax

```
atn2 ( numeric_exp1, numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot ( numeric_exp )
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees ( numeric_exp )
```

radians

Return the degree equivalent of numeric_exp. Results are of the same type as numeric. For expressions of type numeric or decimal, the results have an internal precision of 77 and a scale equal to that of

the numeric expression. When the money datatype is used, internal conversion to float may cause loss of precision.

Syntax

```
radians ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

Report Functions

_add_days

Returns the datetime resulting from adding integer_exp days to timestamp_exp.

Syntax

```
_add_days(timestamp_exp, integer_exp)
```

_add_months

Returns the datetime resulting from adding integer_exp months to timestamp_exp.

Syntax

```
_add_months(timestamp_exp, integer_exp)
```

_add_years

Returns the datetime resulting from adding integer_exp years to timestamp_exp.

Syntax

```
_add_years(timestamp_exp, integer_exp)
```

_age

Returns a number that is obtained from subtracting timestamp_exp from today's date in YYYYMMDD format (years, months, days).

Syntax

```
_age(timestamp_exp)
```

_day_of_week

Returns the day of week (between 1 and 7), where 1 is the first day of the week as indicated by integer_exp (between 1 and 7, 1 being Monday and 7 being Sunday). Note that in ISO 8601 standard, a week begins with Monday being day 1. In North America where Sunday is the first day of the week being day 7.

Syntax

```
_day_of_week(timestamp_exp, integer_exp)
```

Example

```
_day_of_week(2003-01-01, 1) will return 3 because 2003-01-01 was a Wednesday.
```

Result: 3

_day_of_year

Returns the ordinal for the day of the year in date_ exp (1 to 366). Also known as Julian day.

Syntax

```
_day_of_year(timestamp_exp)
```

_days_between

Returns a positive or negative number representing the number of days between the two datetime expressions. If timestamp_exp1 < timestamp_exp2 then the result will be a negative number.

Syntax

```
_days_between(timestamp_exp1, timestamp_exp2)
```

_days_to_end_of_month

Returns a number representing the number of days remaining in the month represented by the datetime expression timestamp_exp.

Syntax

```
_days_to_end_of_month(timestamp_exp)
```

_first_of_month

Returns a datetime that is the first day of the month represented by timestamp_exp.

Syntax

```
_first_of_month(timestamp_exp)
```

_last_of_month

Returns a datetime that is the last day of the month represented by timestamp_exp.

Syntax

```
_last_of_month(timestamp_exp)
```

_make_timestamp

Returns a timestamp constructed from integer_exp1 (the year), integer_exp2 (the month) and integer_exp3 (the day). The time portion defaults to 00:00:00.000 .

Syntax

```
_make_timestamp(integer_exp1, integer_exp2, integer_exp3)
```

_months_between

Returns a positive or negative number representing the number of months between the two datetime expressions. If timestamp_exp1 < timestamp_exp2 then the result will be a negative number.

Syntax

```
_months_between(timestamp_exp1, timestamp_exp2)
```

_week_of_year

Returns the number of the week of the year of the timestamp_exp according to the ISO 8601, in which week 1 of the year is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on Monday (day1) and ends on Sunday (day7).

Syntax

```
_week_of_year(timestamp_exp)
```

_years_between

Returns a positive or negative integer number representing the number of years from timestamp_exp1 to timestamp_exp2. If timestamp_exp1 < timestamp_exp2 then a negative value is returned.

Syntax

```
_years_between(timestamp_exp1, timestamp_exp2)
```

_ymdint_between

Returns a number representing the difference between the datetime expressions timestamp_exp1 and timestamp_exp2. This value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

Syntax

```
_ymdint_between(timestamp_exp1, timestamp_exp2)
```

abs

Returns the absolute value of numeric_exp. The sign of negative values is changed to positive.

Syntax

```
abs(numeric_exp)
```

AsOfDate

Returns the date value of the As Of Time expression, if it is defined. Otherwise, AsOfDate returns the report execution date.

Syntax

```
AsOfDate()
```

AsOfTime

Returns the time value of the As Of Time expression, if it is defined. Otherwise, AsOfTime returns the report execution time.

Syntax

```
AsOfTime()
```

BurstKey

Returns burst key.

Syntax

```
BurstKey()
```

BurstRecipients

Returns the distribution list of burst recipients.

Syntax

```
BurstRecipients()
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling(numeric_exp)
```

CellValue

Returns the value of the current crosstab cell.

Syntax

```
CellValue()
```

character_length

Returns the number of characters in string_exp.

Syntax

```
character_length(string_exp)
```

ColumnNumber

Returns the current column number.

Syntax

```
ColumnNumber()
```

CubeCreatedOn

Returns the date and time when the cube was created. The dimension parameter specifies from which cube to retrieve the metadata. If the dimension source is a Cognos PowerCube (.mdc), the function returns a blank string. The initial creation date of a PowerCube is not maintained.

Syntax

```
CubeCreatedOn(dimension)
```

CubeCurrentPeriod

Returns the current period for the cube. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeCurrentPeriod(dimension)
```

CubeDataUpdatedOn

Returns the date time that data in the cube was last updated. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeDataUpdatedOn(dimension)
```

CubeDefaultMeasure

Returns the name of the default measure for the cube. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeDefaultMeasure(dimension)
```

CubeDescription

Returns the description of the cube. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeDescription(dimension)
```

CubeIsOptimized

Returns true if the cube is optimized. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeIsOptimized(dimension)
```

CubeName

Returns the name of the cube. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeName(dimension)
```

CubeSchemaUpdatedOn

Returns the date time that cube schema was last updated. The dimension parameter specifies from which cube to retrieve the metadata.

Syntax

```
CubeSchemaUpdatedOn(dimension)
```

exp

Returns e raised to the power of numeric_exp. The constant e is the base of the natural logarithm.

Syntax

```
exp(numeric_exp)
```

extract

Returns an integer representing the value of datepart_exp ("year", "month", "day", "hour", "minute", "second" (=default)) in datetime_exp.

Syntax

```
extract(datepart_exp , datetime_exp)
```

floor

Returns the largest integer less than or equal to numeric_exp.

Syntax

```
floor(numeric_exp)
```

GetLocale

Returns run locale (deprecated).

Syntax

```
GetLocale()
```

HorizontalPageCount

Returns current horizontal page count.

Syntax

```
HorizontalPageCount()
```

HorizontalPageNumber

Returns current horizontal page number.

Syntax

```
HorizontalPageNumber()
```

InScope

Returns boolean 1 (TRUE) when cell is in the scope of the data items and MUNs; otherwise, 0 (FALSE).

Syntax

```
InScope(dataItem, MUN, ...)
```

IsBursting

Returns boolean 1 (TRUE) when report will be distributed to given recipient; otherwise, 0 (FALSE) .

Syntax

```
IsBursting('recipientName')
```

IsCrosstabColumnNodeMember

Returns boolean 1 (TRUE) if the current node is a crosstab column node member.

Syntax

```
IsCrosstabColumnNodeMember()
```

IsCrosstabRowNodeMember

Returns boolean 1 (TRUE) if the current node is a crosstab row node member.

Syntax

```
IsCrosstabRowNodeMember()
```

IsFirstColumn

Returns boolean 1 (TRUE) if the current column is the first column.

Syntax

```
IsFirstColumn()
```

IsInnerMostCrosstabColumnNodeMember

Returns boolean 1 (TRUE) if the current node is an innermost crosstab column node member.

Syntax

```
IsInnerMostCrosstabColumnNodeMember()
```

IsInnerMostCrosstabRowNodeMember

Returns boolean 1 (TRUE) if the current node is an innermost crosstab row node member.

Syntax

```
IsInnerMostCrosstabRowNodeMember()
```

IsLastColumn

Returns boolean 1 (TRUE) if the current column is the last column.

Syntax

```
IsLastColumn()
```

IsLastInnerMostCrosstabColumnNodeMember

Returns boolean 1 (TRUE) if the current node is the last innermost crosstab column node member.

Syntax

```
IsLastInnerMostCrosstabColumnNodeMember()
```

IsLastInnerMostCrosstabRowNodeMember

Returns boolean 1 (TRUE) if the current node is the last innermost crosstab row node member.

Syntax

```
IsLastInnerMostCrosstabRowNodeMember()
```

IsOuterMostCrosstabColumnNodeMember

Returns boolean 1 (TRUE) if the current node is an outermost crosstab column node member.

Syntax

```
IsOuterMostCrosstabColumnNodeMember()
```

IsOuterMostCrosstabRowNodeMember

Returns boolean 1 (TRUE) if the current node is an outermost crosstab row node member.

Syntax

```
IsOuterMostCrosstabRowNodeMember()
```

IsPageCountAvailable

Returns boolean 1 (TRUE) if the page count is available for the current execution of the report.

Syntax

```
IsPageCountAvailable()
```

ln

Returns the natural logarithm of the numeric_exp.

Syntax

```
ln(numeric_exp)
```

Locale

Returns run locale.

Syntax

```
Locale()
```

lower

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lower(string_exp)
```

mapNumberToLetter

The user should supply the initial letter and the number is added to it.

Syntax

```
mapNumberToLetter(string_exp, integer_exp)
```

Example

```
mapNumberToLetter ('a', 1) will result in 'b'.
```

Result: `'b'`

mod

Returns the integer value of the remainder (modulo) when integer_exp1 is divided by integer_exp2.

Syntax

```
mod (integer_exp1, integer_exp2)
```

ModelPath

Returns model path.

Syntax

```
ModelPath()
```

Now

Returns current system time.

Syntax

```
Now()
```

nullif

Returns NULL if string_exp1 equals string_exp2 (case insensitive), otherwise returns string_exp1.

Syntax

```
nullif(string_exp1, string_exp2)
```

octet_length

Returns the number of bytes in string_exp.

Syntax

```
octet_length(string_exp)
```

PageCount

Returns the current page count. If you run the report, this function works only when the report output is PDF or Excel. If you save the report output, this function works for all formats.

Syntax

```
PageCount()
```

PageName

Returns the current page name.

Syntax

```
PageName()
```

PageNumber

Returns current page number.

Syntax

```
PageNumber()
```

ParamCount

Returns parameter count of the variable identified by parameterName.

Syntax

```
ParamCount('parameterName')
```

ParamDisplayValue

Returns a string that is the parameter display value of the variable identified by parameterName.

Syntax

```
ParamDisplayValue('parameterName')
```

ParamName

Returns parameter name of the variable identified by parameterName.

Syntax

```
ParamName('parameterName')
```

ParamNames

Returns all parameter names.

Syntax

```
ParamNames()
```

ParamValue

Returns a string that is the parameter value of the variable identified by parameterName.

Syntax

```
ParamValue('parameterName')
```

position

Returns the integer value representing the starting position of string_exp1 in string_exp2 or 0 when string_exp1 is not found.

Syntax

```
position(string_exp1, string_exp2)
```

power

Returns num_exp1 raised to the power of num_exp2.

Syntax

```
power(num_exp1, num_exp2)
```

ReportAuthorLocale

Returns author locale.

Syntax

```
ReportAuthorLocale()
```

ReportCreateDate

Returns the date when the report was created.

Syntax

```
ReportCreateDate()
```

ReportDate

Returns report execution date and time.

Syntax

```
ReportDate()
```

ReportDescription

Returns report description. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportDescription()
```

ReportID

Returns the report id.

Syntax

```
ReportID()
```

ReportLocale

Returns run locale.

Syntax

```
ReportLocale()
```

ReportName

Returns report name. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportName()
```

ReportOption

Returns the value of run option variable identified by optionName. Possible values for optionName: attachmentEncoding, burst, cssURL, email, emailAsAttachment, emailAsURL, emailBody, emailSubject, emailTo, emailToAddress, history, metadataModel, outputEncapsulation, outputFormat, outputLocale, outputPageDefinition, outputPageOrientation, primaryWaitThreshold, print, printer, printerAddress, prompt, promptFormat, saveAs, saveOutput, secondaryWaitThreshold, verticalElements, xslURL.

Syntax

```
ReportOption('optionName')
```

ReportOutput

Returns the name of the output format. Possible return values: CSV, HTML, layoutDataXML, MHT, PDF, rawXML, singleXLS, spreadsheetML, XLS, XML, XLWA.

Syntax

```
ReportOutput()
```

ReportPath

Returns report path. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportPath()
```

ReportProductLocale

Returns product locale.

Syntax

```
ReportProductLocale()
```

ReportSaveDate

Returns the date when the report was last saved.

Syntax

```
ReportSaveDate()
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point.

Syntax

```
round(numeric_exp, integer_exp)
```

Example

```
round (125, -1) rounds to 130.
```

Result: `130.`

RowNumber

Returns current row.

Syntax

```
RowNumber()
```

ServerLocale

Returns the locale of the server that runs the report.

Syntax

```
ServerLocale()
```

ServerName

Returns the name of the server that runs the report.

Syntax

```
ServerName()
```

sqrt

Returns the square root of numeric_exp. numeric_exp must be non-negative.

Syntax

```
sqrt(numeric_exp)
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters or to the end of string_exp if integer_exp2 is -1. The first character in string_exp is at position 1.

Syntax

```
substring(string_exp , integer_exp1, integer_exp2)
```

TOCHeadingCount

Returns the table of contents heading count for a specific heading level.

Syntax

```
TOCHeadingCount(headingLevel)
```

Today

Returns current system date.

Syntax

```
Today()
```

trim

Returns a string_exp trimmed of leading and/or trailing blanks or trimmed of a character specified in match_character_exp. trim_what_exp may be: "LEADING", "TRAILING" or "BOTH" (default) . match_character_exp = empty string to trim blanks, or specifies a character to be trimmed. String_exp = the string to be trimmed.

Syntax

```
trim(trim_what_exp, match_character_exp, string_exp)
```

upper

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
upper(string_exp)
```

URLEncode

Returns the url encoded value of the input text.

Syntax

```
URLEncode('text')
```

Data Type Casting Functions

date2string

Returns a date as string in YYYY-MM-DD format.

Syntax

```
date2string(date_exp)
```

date2timestamp

Converts a date to a timestamp. The time part of the timestamp will be zero.

Syntax

```
date2timestamp(date_exp)
```

date2timestampTZ

Converts a date to a timestamp with time zone. The time and time zone parts of the timestamp will be zero.

Syntax

```
date2timestampTZ(date_exp)
```

DTinterval2string

Returns a date time interval as string in "DDDD HH:MM:SS.FFFFFFF" or "-DDDD HH:MM:SS.FFF" format.

Syntax

```
DTinterval2string(DTinterval_exp)
```

DTinterval2stringAsTime

Returns a date time interval as string. The days are converted to hours. Output format: "HHHH:MM:SS.FFFFFFF" or "HH:MM:SS.FFF".

Syntax

```
DTinterval2stringAsTime(DTinterval_exp)
```

int2DTinterval

Converts an integer to a date time interval. String_exp specifies what the integer_exp is representing: "ns"=nanoseconds, "s"=seconds (=default), "m"=minutes, "h"=hours, "d"=days.

Syntax

```
int2DTinterval(integer_exp, string_exp)
```

int2YMinterval

Converts an integer to a year month interval. String_exp specifies what the integer_exp is representing: "y"=years, "m"=months (=default).

Syntax

```
int2YMinterval(integer_exp, string_exp)
```

number2string

Converts a number to a string, using the %g format specifier (C/C++ syntax).

Syntax

```
number2string(num_exp)
```

string2date

Returns a date string as date in "YYYY-MM-DD" format.

Syntax

```
string2date(string_exp)
```

string2double

Returns a floating point number. The string argument to the function "string2double" has the following form:"[whitespace] [sign] [digits] [digits] [{d | D |e | E }[sign]digits]"

Syntax

```
string2double(string_exp)
```

string2DTinterval

Returns a date-time interval string as Date Time Interval in "[-]DD HH:MM[:SS[.FFF]]" format.

Syntax

```
string2DTinterval(string_exp)
```

string2int32

Returns an integer. The string argument to the function "string2int32" has the following form: "[whitespace] [{+ | -}] [digits]"

Syntax

```
string2int32(string_exp)
```

string2int64

Returns a long integer. The string argument to the function "string2int64" has the following form: "[whitespace] [{+ | -}] [digits]"

Syntax

```
string2int64(string_exp)
```

string2time

Returns a time string as time in "HH:MM:SS.FFFFFFF" format.

Syntax

```
string2time(string_exp)
```

string2timestamp

Returns a timestamp string as timestamp in "YYYY-MM-DD [white space]+ HH:MM:SS.FFFFFFF" format.

Syntax

```
string2timestamp(string_exp)
```

string2timestampTZ

Returns a timestamp with time zone "YYYY-MM-DD HH:MM:SS.FFFFFFF +HHMM" or "YYYY-MM-DD HH:MM:SS.FFF -HHMM" format.

Syntax

```
string2timestampTZ(string_exp)
```

string2YMinterval

Returns a year-month interval string as Year Month Interval in "[-]YY MM" format.

Syntax

```
string2YMinterval(string_exp)
```

time2string

Returns a time as string in HH:MM:SS.FFF format.

Syntax

```
time2string(time_exp)
```

timestamp2date

Converts a timestamp to a date. The time part of the timestamp will be ignored.

Syntax

```
timestamp2date(timestamp_exp)
```

timestamp2string

Returns a timestamp as string in YYYY-MM-DD HH:MM:SS.FFFFFFF format.

Syntax

```
timestamp2string(timestamp_exp)
```

timestamp2timestampTZ

Converts a timestamp to a timestamp with time zone. The displacement part of the timestamp with time zone will be zero.

Syntax

```
timestamp2timestampTZ(timestamp_exp)
```

timestampTZ2date

Converts a timestamp with time zone to a date. The time and time zone parts of the timestamp will be ignored.

Syntax

```
timestampTZ2date(timestampTZ_exp)
```

timestampTZ2string

Returns a timestamp with time zone as string in "YYYY-MM-DD HH:MM:SS.FFFFFFF +HHMM" or "YYYY-MM-DD HH:MM:SS.FFF -HHMM" format.

Syntax

```
timestampTZ2string(timestampTZ_exp)
```

timestampTZ2timestamp

Converts a timestamp with time zone to a timestamp. The displacement part of the timestamp with time zone will be ignored.

Syntax

```
timestampTZ2timestamp(timestampTZ_exp)
```

timeTZ2string

Returns a time with time zone as string in "HH:MM:SS.FFF +HHMM" or "HH:MM:SS.FFFFFF -HHMM" format.

Syntax

```
timeTZ2string(timeTZ_exp)
```

Example

`"-05:30" means a TimeZone of GMT minus 5 hours and 30 minutes.`

Result: `GMT minus 5 hours and 30 minutes`

YMinterval2string

Returns a year month interval as string in "(YY MM)" or "-(YY MM)" format.

Syntax

```
YMinterval2string(YMinterval_exp)
```

Appendix F: Producing Reports in Microsoft Excel Format

There are limitations when producing reports in Microsoft Excel format, some of them originating with Microsoft Excel and some of them with Cognos 8.

Unable to Load Images from the Cognos 8 Content Store in a Report

If a report contains an image whose URL points to the Cognos 8 content store, Microsoft Excel generates an access violation error and shuts down.

This problem is a known issue in the Microsoft knowledge base, and Microsoft is currently investigating the problem. This problem occurs only in Excel 2000 and 2002.

A Blank Worksheet is Opened

If Microsoft Excel cannot download a worksheet within a timeout period, Excel may instead open a blank worksheet.

A Warning Message Appears When Excel Opens a Cognos 8 Report

Each time Microsoft Excel opens a Cognos 8 report, the following message appears:

Some of the files in this Web page aren't in the expected location. Do you want to download them anyway? If you're sure the Web page is from a trusted source, click Yes.

The Excel workbook in HTML/XML format requires the presence of the file filelist.xml. Cognos 8 does not allow the creation of local files on the client side. In addition, a local file that contains URLs introduces a security issue. Consequently, this message will appear whenever you open a Cognos 8 report in Excel.

Spreadsheet Content Not Saved for Reports Saved in XLS Format

If you open a report that was saved in XLS format or run a report in XLS format, and security settings in your browser are set so that you are prompted to open or save the report, do not click **Save**. If you save the report, the spreadsheet content will not be saved. This is because Excel reports in Office 2000 HTML format use relative paths to the spreadsheets. The relative URL paths are no longer available when you open a saved XLS report.

Instead, click **Open** first and then choose to save the report.

Unable to Load Excel Report in Netscape 7.01

This version of Cognos 8 does not support loading Microsoft Excel reports in Netscape 7.01.

Unable to Nest Labels in Charts

Currently, it is not possible to specify nested labels for the category axis via XML.

Data Series Are Truncated

Microsoft Excel may group data series or categories differently when compared to a chart produced by Cognos 8.

A 1 KB buffer limit in Excel limits the maximum number of data series per chart to 120. Data series over 120 are truncated.

Colors in Excel Different From Those in HTML or PDF

If a Cognos 8 report contains a chart that uses custom colors, Excel may not be able to add the custom color to the Excel color palette. Excel will attempt to match the custom color to one of its available standard colors. As a result, colors in Excel may vary from those seen in HTML or PDF.

We recommend that you use standard colors in charts.

Repeating Pie Charts

If you have a report that has repeating pie charts, and you define a chart title, Excel will show each pie with a title that is a concatenation of the chart title and the data series. For example, if the chart title is Quantity Sold by Order Method and Product Line and the data series is Order method, the title of each pie in Excel will be

Quantity Sold by Order Method and Product Line, *order method*

Unable to Skip Discrete Axis Labels

In Cognos 8 charts, you can control the skipping of discrete axis labels. This feature is not supported in Excel charts.

Unsupported Cognos 8 Formatting

About 30% of the formatting functions available in Cognos 8 are not supported in Microsoft Excel. In particular, Excel does not allow changing locale-dependent formatting attributes, such as the following:

- Decimal Separator
- Exponential Symbol
- Group Separator
- Monetary Decimal Separator
- AM String
- Day Name
- Day Short Name
- Decimal Delimiter Symbol
- Month Name
- Month Short Name
- PM String

In addition, Excel does not support the following

- Format Width
- International Currency Symbol
- List Separator
- Percent Symbol (Excel does not support percent symbols for charts.)
- Multiplier
- Overline Text Format
- PerMill Symbol
- Plus Sign
- Scale (Excel has a different scaling formula than Cognos 8.)
- Calendar (Excel does not allow changing the calendar.)
- Era Name

- First Day Of Week
- Show Era

Cell Contains Series of

Cells in Microsoft Excel have a limit of 255 characters. If your report contains text strings that are longer than 255 characters, they will be formatted as text and appear as ######.

To resolve this problem, use fewer characters.

Reports with More Than 256 Columns Cannot Be Rendered in Excel

Microsoft Excel limits the size of a worksheet size to 65536 rows by 256 columns. If your report contains more than 65536 rows, it is split into multiple worksheets. The number of worksheets that your report can contain is limited by the physical memory of your computer. If your report contains more than 256 columns, the following error occurs:

Reports with more than 256 columns cannot be rendered in Excel.

Table and Column Width

Microsoft Excel does not support using percentages to determine the width of tables. If the report contains only one table, the value of the width attribute for the Table element in the report specification determines the width of the table in the Excel worksheet. If the report contains more than one table, Excel determines the width of all the tables in the worksheet. If the tables are nested, the width specified for the outer table is used and, if necessary, the width is adjusted to accommodate data in the nested tables. The columns and rows around the table are merged to preserve the appearance of the nested table. When the workbook is saved, only a single table is saved per worksheet.

Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions

SSL is supported for only the following formats and Microsoft Excel versions.

Format	Version
Excel 2000 single sheet	Microsoft Excel 2000, 2002, XP, 2003
Excel 2000	Microsoft Excel 2003

Format	Version
Excel 2002	Microsoft Excel 2002, XP, 2003

Number Format Becomes Currency Format in Japanese Excel

A report uses the Number data format and you save it as Microsoft Excel output. When you open the report in the Japanese version of Microsoft Excel, the data format is listed as Currency rather than Number. This occurs because Japanese Excel interprets the standard Number data format slightly differently than other versions of Excel.

The value appears correctly in Number format. For example, if you specified five digits as your number format, five digits still appear. In Excel, click the **Custom** number format to see the exact format string being used.

Report Shows Data in Wrong Columns

A report contains a large amount of data that is presented using a large number of nested report objects, such as tables and blocks. When the report is produced in Excel format, some of the data appears in the wrong columns. This occurs because Excel has a 64K limit on how many nested cell objects can appear in a single spreadsheet.

Avoid exceeding the Excel limit by redesigning the report to present the data using non-nested structures.

Unable to Access a Report on a Remote Server

You cannot access a report in Excel format on a remote server.

To resolve this problem, you must change the hostname portion of the gateway URI from localhost to either the IP address of the computer or the computer name. You do this using Cognos Configuration.

Drill-through Reports Are Not Supported

Cognos 8 does not support drill-through for reports in Excel format.

Map Reports Are Not Supported

Cognos 8 does not support map reports in Excel format.

Unsupported Excel Formatting

Cognos 8 does not support the following formatting functions available in Excel:

- background images in table cells
- Excel-specific headers and footers
- text flow and justification
- floating text objects
- white space, normal, and wrap text formatting
- maximum characters

Some layouts do not show exactly in HTML and PDF due to Microsoft Excel limitations.

Hyperlink Buttons Are Not Supported

Microsoft Excel does not support hyperlink buttons.

Unable to View Reports in Excel Format Sent as Email Attachments

Cognos 8 can send Excel reports in HTML and XML format by email. However, you cannot open them directly from the email message.

Save the Excel email attachments to your computer and view them from there.

Unsupported Chart Properties in Excel

The following Cognos 8 chart properties are not supported in Microsoft Excel:

- tool tips
- conditional text
- depth
- visual angle
- show values
- marker text location
- show baseline
- new note
- new marker

- truncation text and allow n-degrees rotation category labels
- border
- margin
- box type
- font and font alignment
- footer
- subtitle
- regression line
- baseline

In addition, Cognos 8 makes sure that Excel reuses the same color palette that is defined in Cognos 8. However, Excel can only use the first 16 colors from the Cognos 8 palette. If the number of categories in a chart exceeds 16, the rest of the colors are taken from the default Excel palette.

Unsupported Chart Types in Excel

About 30% of the chart types available in Cognos 8 are not matched in Microsoft Excel. The following table shows which chart types are not supported in Microsoft Excel 2000 and later. Charts that are not supported appear as a default column chart in Microsoft Excel.

The following chart types appear differently or are not supported:

- Bubble charts

  Excel does not support regression lines.

- Combination charts

  In Excel, they appear as two-dimensional combination charts. If a 3-D or two-dimensional combination chart include only an area, column, or line chart, only the one chart appears.

- Gauge charts
- Donut charts

  Excel may fill in the donut hole to accommodate extra measures and shows donut 3-D charts as donut charts.

- Maps

  In Excel, no chart appears.

- Metrics range charts

  In Excel, they appear as combination charts with two line charts for the tolerance bars.

- Pareto charts

  In Excel, the cumulative line is not displayed.

- Pie charts

  Excel shows only one type of pie chart and may show it at a different angle.

- Progressive column charts
- Polar charts

  In Excel, a scatter chart appears instead.

- Quadrant charts
- Radar charts

  Excel does not support stacked area radar charts. Excel names area radar charts as filled radar charts. Excel shows radar charts as radar with markers.

- Scatter charts

  In Excel, 3-D scatter charts appears as two-dimensional scatter charts. The Z axis is dropped.

Appendix G: Report Studio Object and Property Reference

This appendix contains definitions of the objects and properties found in Report Studio. They are available contextually, by pressing F1 when an object or property is active in the Report Studio authoring environment.

The following objects and properties are referenced:

- Report Studio Objects

  These objects are visible in the Report Studio work area. They can be inserted from the **Toolbox** tab .

- Report Studio Properties

  These properties can be set on Report Studio objects. They are available in the lower-left pane of Report Studio.

- Data Formatting Properties

  These properties can be set on data values by selecting **Layout Data Format** from the **Data** menu, or by editing the **Data Format** property for Report Studio objects.

Report Studio Objects

The following is a list of objects available in Report Studio.

3-D Area

A chart in which members of a data series are represented by three-dimensional areas of varying size and color. The three-dimensional area chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Area

Border Color, Borders, Chart Type, Values

3-D Bar

A chart in which members of a data series are represented by three-dimensional bars of varying length and color. The three-dimensional bar chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Bar

Border Color, Borders, Chart Type, Values

3-D Combination Chart

A chart that includes any number and combination of three-dimensional bar charts, line charts, and area charts. This chart plots any number of data series against one common measure on the vertical numeric axis and one or two common data series on ordinal axes.

Properties of 3-D Combination Chart

3-D Viewing Angle, Background Color, Background Image, Baselines, Border, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Series Color, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible, X Axis, Y Axis

3-D Line

A chart in which members of a data series are represented by three-dimensional lines of varying colors. The three-dimensional line chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Line

Border Color, Borders, Chart Type, Values

3-D Scatter Chart

A chart that plots three measures against one or more data series on a three-dimensional X-Y-Z graph.

Properties of 3-D Scatter Chart

3-D Viewing Angle, Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Relative Alignment, Render Page when Empty, Render Variable, Show Feelers, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Values, Visible

Angular Axis

The angular numeric axis for a polar chart, including labels, titles, range, and scale.

Properties of Angular Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Angular Measure

The measure that determines the angular position of each data marker on a polar chart.

Properties of Angular Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Area

The data marker used to represent data series in an area chart.

Properties of Area

Axis Assignment, Border Color, Borders, Chart Type, Grouping Type, Value Location, Values, Value Type

As of Time Expression

An expression that produces a Date-Time value. This expression can be used to show report results for a specific time period that is defined by an expression that you create.

Properties of As of Time Expression

Report Expression

Axis Title

The title for an axis of the chart.

Properties of Axis Title

Box Type, Classes, Conditional Styles, Default Title, Font, Foreground Color, Master Detail Relationships, Properties, Query, Style Variable, Visible

Bar

A chart in which members of a data series are represented by bars of varying length and color.

Properties of Bar

Axis Assignment, Border Color, Borders, Chart Type, Connecting Lines, Grouping Type, Value Location, Values, Value Type

Baseline

A baseline to be rendered on a chart.

Properties of Baseline

Aggregate Function, Axis Assignment, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Hierarchy, Label, Label, Legend Label, Level, Line Styles, Master Detail Relationships, Member, Member Offset (%), Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Query, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

Baseline

A baseline for a polar chart, scatter chart, or bubble chart.

Properties of Baseline

Aggregate Function, Axis Assignment, Data Item Value, Expression, Hierarchy, Label, Label, Legend Label, Level, Line Styles, Master Detail Relationships, Member, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

Baseline

A baseline for a three-dimensional combination chart.

Properties of Baseline

Aggregate Function, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Hierarchy, Label, Level, Line Styles, Master Detail Relationships, Member, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

Baseline

A baseline for a three-dimensional scatter chart.

Properties of Baseline

Aggregate Function, Box Type, Data Item Value, Expression, Hierarchy, Label, Level, Line Styles, Master Detail Relationships, Member, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

Block

A container into which you can insert other objects.

Properties of Block

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

Bookmark

A link to another area within the same report. The link can be defined as a static value, a query item, or as the result of a report expression.

Properties of Bookmark

Data Item Label, Data Item Value, Label, Report Expression, Source Type

Bubble Chart

A point chart that plots one or more data series against three measures: a Y-value, an X-value, and a bubble whose relative size represents the third measure. Multiple points are plotted for each category.

To help distinguish values, set the Tool Tips property of this object to Yes.

Properties of Bubble Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Query, Regression Line, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Bubble Measure

The measure that determines the size of each bubble on a bubble chart.

Properties of Bubble Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Calculated Measure

A data item that is a calculated member.

Properties of Calculated Measure

Calculation Intersection, Dimension, Expression, Label, Name, Solve Order, Type

Calculated Member

An item, within a dimension, that represents an occurrence of schema data defined as a calculation of two or more members.

Properties of Calculated Member

Caption, Data Item, Name

Calculated Member

A data item that is a calculated member.

Properties of Calculated Member

Calculation Intersection, Dimension, Expression, Hierarchy, Label, Name, Solve Order, Type

Caption

The caption on a Field Set object.

Properties of Caption

Background Color, Background Image, Border, Classes, Conditional Styles, Font, Foreground Color, Margin, Padding, Style Variable

Chart Body

Defines the body style of the chart itself. The style of the body can be set independently of the chart itself.

Properties of Chart Body

Background Color, Background Image, Conditional Styles, Fill Effects, Font, Foreground Color, Horizontal Alignment, Style Variable

Chart Footer

A footer for the chart.

Properties of Chart Footer

Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Chart Node Member

A data item, and its accompanying text, to render on the chart.

Properties of Chart Node Member

Aggregate Function, Custom Label, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Hierarchy, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Properties, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Sorting, Source Type, Source Type, Source Type, Source Type, Text, Text Source Variable, Type, URL, URL Source Variable

Chart Subtitle

The subtitle for a chart.

Properties of Chart Subtitle

Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Chart Text Item

The data source and format for a text item, such as a legend item, legend title, axis label, or axis title.

Properties of Chart Text Item

Aggregate Function, Conditional Styles, Data Format, Data Item Label, Data Item Value, Expression, Hierarchy, Label, Level, Member, Name, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Style Variable, Text, Text Source Variable, Type

Chart Title

The title text that appears at the top of the chart.

Properties of Chart Title

Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Class

The HTML class name for a layout object. Use this attribute to indicate the type of styling to apply to the object when the report is rendered.

Properties of Class

Background Color, Background Image, Border, Box Type, Description, Floating, Font, Foreground Color, Horizontal Alignment, Label, Label, Margin, Padding, Relative Alignment, Size & Overflow, Spacing & Breaking, Table Properties, Text Flow & Justification, Vertical Alignment, Visible, White Space

Combination Chart

A chart that uses combinations of column charts, area charts, and line charts as data markers to plot multiple data series.

Properties of Combination Chart

Background Color, Background Image, Baselines, Border, Box Type, Chart Orientation, Classes, Conditional Palette, Conditional Styles, Depth, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Ordinal Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Rotate Values, Series Color, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Visible, Visual Angle, Y1 Axis, Y2 Axis, Y2 Axis Position

Component Override

Overrides a child object of the Layout Component Reference object.

Properties of Component Override

Component Reference

Conditional Block

A block that can be used for conditional display.

Properties of Conditional Block

Background Color, Background Image, Block Variable, Border, Box Type, Classes, Conditional Styles, Current Block, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

Conditional Block

Contains the default set of layout objects to render based on a report variable.

Properties of Conditional Block

Background Color, Background Image, Block Variable, Border, Box Type, Classes, Conditional Styles, Current Block, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Padding,

Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

Context Item Text

The text associated with a context area item.

Properties of Context Item Text

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Margin, Padding, Relative Alignment, Render Variable, Separator, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible

Crosstab

A layout object used to render the results of a query that aggregates data, and then arranges it in a two-dimensional grid.

Properties of Crosstab

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Default Measure, Fact Cells Precedence, Floating, Font, Foreground Color, Has Fact Cells, Horizontal Alignment, Margin, Master Detail Relationships, Name, No Data Contents, Pagination, Query, Relative Alignment, Render Page when Empty, Render Variable, Rows Per Page, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Crosstab Columns

Overrides the style for Crosstab Column Member objects that is defined in the GlobalReportStyles.css file.

Properties of Crosstab Columns

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Columns

A list of columns in a crosstab.

Properties of Crosstab Columns

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Corner

The top-left corner of a crosstab, on top of the row labels and to the left of the column labels. It is generally used to represent crosstab members.

Properties of Crosstab Corner

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Fact Cells

The contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

Properties of Crosstab Fact Cells

Aggregate Function, Apply Single Class, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Intersection

An intersection in a crosstab. The cell contents of a specific intersection can be overridden and the style defined.

Properties of Crosstab Intersection

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Column Coordinate, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Row Coordinate, Set Sorting, Size & Overflow, Source Type, Source Type,

Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Member Fact Cells

The contents of the fact cells of a crosstab node member.

Properties of Crosstab Member Fact Cells

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Node Member

A member in the crosstab node.

Properties of Crosstab Node Member

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Indentation, Member, Name, Node Coordinate, Padding, Pagination, Properties, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Sorting, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Rows

Overrides the style for Crosstab Row Member objects that is defined in the GlobalReportStyles.css file.

Properties of Crosstab Rows

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Rows

A list of rows in a crosstab.

Properties of Crosstab Rows

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Space

Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge.

Properties of Crosstab Space

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Node Coordinate, Padding, Pagination, Render Fact Cells, Report Expression, Report Expression, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

Cumulation Line

A line that shows the cumulative effect of multiple series members on a measure in a pareto chart.

Properties of Cumulation Line

Cumulation Axis, Cumulation Label, Line Styles, Point Shape, Point Size (pt), Properties, Value Location, Values

Cumulation Line Axis

The axis for the cumulation line in a pareto chart.

Properties of Cumulation Line Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Cumulation Line Label

A label that is rendered with the cumulation line in a pareto chart.

Properties of Cumulation Line Label

Aggregate Function, Data Item Label, Data Item Value, Expression, Hierarchy, Label, Level, Member, Name, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Text, Text Source Variable, Type

Data Item

A set of data values or members.

Properties of Data Item

Aggregate Function, Calculation Intersection, Detail, Expression, Label, Name, Pre-Sort, Rollup Aggregate Function, Solve Order, Type

Date

The date when the report runs.

Properties of Date

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

Date & Time Prompt

A prompt control with which you can select a date and time value.

Properties of Date & Time Prompt

Box Type, Calendar Type, Clock Mode, Conditional Styles, Default Selections, Display Milliseconds, Display Seconds, First Date, Floating, Hide Adornments, Last Date, Multi-Select, Name, Parameter, Range, Render Variable, Required, Select UI, Style Variable, Visible

Date Prompt

A prompt control with which you can to select a date value.

Properties of Date Prompt

Box Type, Calendar Type, Conditional Styles, Default Selections, First Date, Floating, Hide Adornments, Last Date, Multi-Select, Name, Parameter, Range, Relative Alignment, Render Variable, Required, Select UI, Style Variable, Visible

Default Measure

A default measure for the chart. If the chart measure cannot be determined by the data series rendered on the chart edges, the default measure is used.

Properties of Default Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Detail Filter

A set of conditions in a query that narrow the scope of the data returned. A detail filter is applied before aggregation is complete.

Properties of Detail Filter

Application, Expression, Usage

Dimension

A grouping of descriptive information about an aspect of a business. Dimensions contain levels, whose order defines the hierarchy of organizational structures and data. Dimensions and levels are values by which measures can be viewed, filtered, or aggregated.

Properties of Dimension

Name

Display Layer

A map layer that is there for appearance only. Display layers do not correspond to data series or measures.

Properties of Display Layer

Border Color, Borders, Conditional Styles, Fill Effects, Labels, Style Variable

Explicit Member Set

A set of data items that define an explicit set of members.

Properties of Explicit Member Set

Dimension, Hierarchy, Label, Name, Set Sorting, Type

Fact

The central values that are aggregated and analyzed. Also known as measures, they are special business measurement values, such as sales or inventory levels.

Properties of Fact

Data Item, Name

Field Set

A container with a caption, into which you can insert other objects. It is similar to a block object, except that it also has a caption.

Properties of Field Set

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Relative Alignment, Render Variable, Show Caption, Size & Overflow, Style Variable, Visible

Gauge Chart

A chart that plots a data series against a measure using a dial or gauge for the measure, and needles or indicators for the series members.

Properties of Gauge Chart

Axis Title, Background Color, Background Image, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Dial Outline Color, Drill-Through Definitions, Face Color, Fill Effects, Floating, Font, Footer, Foreground Color, Gauge Labels, Gauge Palette, Legend, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible

Gauge Labels

A label for each gauge in a multiple gauge chart.

Properties of Gauge Labels

Classes, Conditional Styles, Drill-Through Definitions, Font, Foreground Color, Horizontal Alignment, Maximum Truncation Characters, Style Variable, Truncation, Truncation Text, Visible

Gauge Numerical Axis

The numeric axis for the gauge chart, including labels, titles, range, and scale.

Properties of Gauge Numerical Axis

Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Generated Prompt

A control that acts as a placeholder. The report server will replace this control with an appropriate generated prompt control, as if it was on a generated prompt page.

Properties of Generated Prompt

Hide Adornments, Name, Parameter, Render Variable, Required

Hierarchy Set

A set of data items that define the members of a hierarchy.

Properties of Hierarchy Set

Dimension, Hierarchy, Label, Name, Root Members Only, Set Sorting, Type

HTML Item

A container into which you can add HTML, such as a link to a multimedia file. HTML items will only appear when you run the report in HTML format.

When you upgrade to the next version of Cognos 8, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features such as JavaScript that refers to Cognos HTML items.

Properties of HTML Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, Hierarchy, HTML, HTML Source Variable, Label, Level, Member, Name, Name, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Type

Hyperlink

A hyperlink that can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.

Properties of Hyperlink

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Render Variable, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Visible

Hyperlink Button

A hyperlink that is formatted as a button. The hyperlink can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.

Properties of Hyperlink Button

Background Color, Background Image, Border, Classes, Conditional Styles, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Floating, Foreground Color, Margin, Name, Padding,

Relative Alignment, Render Variable, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Style Variable, Text, Text Source Variable, URL, URL Source Variable

Image

A link to an image file. The link can be a static value, or it can come from a report expression or query item. Use the URL source properties of the image object to define the link.

Properties of Image

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Item Label, Data Item Value, Drill-Through Definitions, Expression, Floating, Hierarchy, Label, Level, Margin, Member, Name, Name, Relative Alignment, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Style Variable, Type, URL, URL Source Variable, Visible

Intersection (Tuple)

A data item obtained from the combination of two or more members that you specify. An intersection appears as a single, unified member instead of its component members. Each member must be from a different dimension.

Properties of Intersection (Tuple)

Dimension, Hierarchy, Label, Members, Name, Type

Interval Prompt

An advanced prompt control that allows you to enter time duration values.

Properties of Interval Prompt

Box Type, Conditional Styles, Default Selections, Display Milliseconds, Display Seconds, Floating, Hide Adornments, Multi-Select, Name, Parameter, Range, Render Variable, Required, Style Variable, Visible

Join

A relationship between a field in one table or query and a field of the same data type in another table or query.

Properties of Join

Join Relationships

Key

An object that uniquely identifies members of a level.

If the unique identifier is a primary or alternate key, you need only one key object. If the unique identifier is a composite key, you need one key object for every data item that participates in making the members of a level unique.

Properties of Key

Data Item, Name

Layout Component Reference

A reference to another layout object. Before you can reference an object, its ID property must be set.

Properties of Layout Component Reference

Component Reference, Embed, Overrides

Legend

A key to the patterns or colors assigned to the data series in a chart.

Properties of Legend

Absolute Position, Auto Truncation, Background Color, Background Image, Border Color, Borders, Bottom Position (px), Box Type, Classes, Conditional Styles, Drill-Through Definitions, Fill Effects, Font, Font Auto-Sizing, Foreground Color, Horizontal Alignment, Left Position (px), Legend Title, Maximum Characters, Position, Right Position (px), Separator, Show Legend Values, Style Variable, Top Position (px), Truncation Text, Visible

Legend Title

The title for the legend, including the title text and text style. If this object is empty, a default title is rendered, if available.

Properties of Legend Title

Box Type, Classes, Conditional Styles, Default Title, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Level

A set of members with a predefined set of similar characteristics. For example, the members Year 1999 and Year 2000 in the Time dimension form a year level, while the corresponding quarters form a quarter level.

Properties of Level

Caption, Name, Sorting

Level Hierarchy

Specifies how the levels in a dimension are logically ordered.

Properties of Level Hierarchy

Name

Level Set

A set of data items that define the members of a level.

Properties of Level Set

Dimension, Hierarchy, Label, Level, Name, Set Sorting, Type

Line

The data marker used to represent a data series in a line chart.

Properties of Line

Axis Assignment, Border Color, Borders, Chart Type, Grouping Type, Line, Line Style, Line Weight (pt), Point Shape, Point Size (pt), Value Location, Values, Value Type

List

A layout object that is used to present query results in a list fashion.

Properties of List

Background Color, Background Image, Border, Box Type, Classes, Column Titles, Conditional Styles, Contents Height, Floating, Font, Foreground Color, Grouping & Sorting, Horizontal Alignment, Margin, Master Detail Relationships, Name, No Data Contents, Pagination, Properties, Query, Relative Alignment, Render Page when Empty, Render Variable, Rows Per Page, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

List Cell

A cell in a row, you can use for a list header or footer.

Properties of List Cell

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow

& Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

List Column

A column in a list. Generally, the column will contain a query item, but it may also contain any number of layout objects. For example, the To column in a statement list may present the address in a table format. If the column only contains a query items, then the column will automatically span the group if the query item is grouped.

Properties of List Column

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Pagination, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Column Body

The contents of a column in a list report.

Properties of List Column Body

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Group Span, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

List Columns

A set of columns in a list.

Properties of List Columns

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Columns

Overrides the style for List Column objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Columns Body Style

Overrides the style for List Column Body objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns Body Style

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Columns Title Style

Overrides the style for List Column Title objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns Title Style

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Column Title

The title of a list column.

Properties of List Column Title

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Member, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

List Footer

The footer that appears at the end of a list on each page on which the list is rendered. It is useful for presenting page totals.

Properties of List Footer

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Properties, Push To Bottom, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Header

The header that appears at the start of a list on each page that the list is rendered. It is useful for presenting carry forward totals.

Properties of List Header

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Pagination, Properties, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Page Footer

The footer that appears at the bottom of every page of a list report. This object is rendered after the list details and other list footers.

Properties of List Page Footer

Push To Bottom

List Page Header

The header in the list that will appear on every page rendered. It occurs after the column titles and before the overall group header of the list.

Properties of List Page Header

Display After Overall Header

List Row

A row in a list.

Properties of List Row

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Row Cells Style

Overrides the style for Row Cells Style objects that is defined in the GlobalReportStyles.css file.

Properties of List Row Cells Style

Background Color, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Map

A chart that uses a map to show data.

Properties of Map

Axis Title, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Language, Dictionary, Drill-Through Definitions, Expand Features, Fill Effects, Floating, Font, Footer, Foreground Color, Ignore Data with No Features, Legend, Map & Layers, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, No Data Features Size (pt), Notes, Padding, Pagination, Query, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible

Map Location

Associates a data series with regions on the region layer of the map.

Properties of Map Location

Aggregate Function, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

Map Location

Associates a data series with points on the point layer of the map.

Properties of Map Location

Aggregate Function, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

Map Refinement Location

Qualifies the data series members that are associated with regions on the map.

Properties of Map Refinement Location

Aggregate Function, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

Map Refinement Location

Qualifies the data series members that are associated with points on the map.

Properties of Map Refinement Location

Aggregate Function, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

Marker

An additional marker that can be placed at a static point on a chart.

Properties of Marker

Aggregate Function, Axis Assignment, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Hierarchy, Label, Label, Level, Master Detail Relationships, Member, Name, Numeric Value, Percentile, Percent of Axis, Point Color, Point Shape, Point Size (pt), Position type, Properties, Query, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

Marker

A marker on a point chart. Point charts include scatter charts, polar charts, radar charts, and bubble charts.

Properties of Marker

Box Type, Data Item Value, Expression, Label, Master Detail Relationships, Numeric Value, Percentile, Percent of Axis, Point Color, Point Shape, Point Size (pt), Position type, Position type, Properties, Query, Report Expression, Standard Deviations

MDX

A multidimensional expression (MDX) query against an OLAP data source.

Properties of MDX

Catalog, Data Source, MDX, Name

Measure

A data item that defines a measure.

Properties of Measure

Dimension, Label, Member, Name, Type

Member

A data item that defines a member.

Properties of Member

Dimension, Hierarchy, Label, Member, Name, Type

Member Children Set

A set of data items that define the children set of a member.

Properties of Member Children Set

Dimension, Hierarchy, Label, Member, Name, Set Sorting, Type

Member Hierarchy

The organization of a dimension's members into a logical tree structure, with each member having one or more "parent" members and an arbitrary number of "child" members.

Properties of Member Hierarchy

Name

Member Property

A property that is associated with members of a level. Attributes can be used to refine a search within level members, or to provide additional information about members.

Properties of Member Property

Data Item, Name

Member Property

Properties of Member Property

Dimension, Hierarchy, Label, Level, Name, Property, Type

Member Set

A data item that represents a named set. Member sets supply the context for expressions that require evaluation for multiple dimensions. For example, they determine the contexts for which you can apply analytical functions such as rank.

Properties of Member Set

Data Item, Name

Metrics Range Chart

A chart that superimposes target value markers, target range markers, and tolerance range markers over any number or combination of bar, line, and area charts.

Properties of Metrics Range Chart

Background Color, Background Image, Baselines, Border, Box Type, Classes, Conditional Palette, Conditional Styles, Depth, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Marker Color, Marker Label, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Ordinal Axis, Padding, Pagination, Palette, Performance Pattern, Query, Range Label, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Target Color, Target Marker, Target Marker Border Color, Target Marker Position, Target Range (%), Title, Tolerance Color, Tolerance Label, Tooltips, Upper Range Skew (%), Value Location, Visible, Visual Angle, Y1 Axis

Metric Studio Diagram

A Metric Studio history chart rendered as an image.

Properties of Metric Studio Diagram

Description, Diagram Identifier

Named Set

A reference to a named set from the model.

Properties of Named Set

Dimension, Hierarchy, Label, Name, Set Sorting, Type

Note

A note on a chart. The source of the note text can be static text, a query item, or a report expression.

Properties of Note

Bottom Position (px), Height (px), Left Position (px), Note Border, Width (px)

Note Content

The content and style of a note.

Properties of Note Content

Background Color, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Numerical Axis

The numeric axis for a progressive chart, including labels, titles, range, scale, and gridlines.

Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Numerical Axis

The numeric axis for a pareto chart, including labels, titles, and gridlines.

Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Ordinal Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of Ordinal Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Display Frequency, Drill-Through Definitions, First Label Index, Font, Foreground Color, Gridlines, Horizontal Alignment, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

Page

A page in a layout.

Properties of Page

Background Color, Background Image, Border, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Margin, Master Detail Relationships, Name, Pagination, PDF Page Setup, Properties, Query, Render Variable, Size & Overflow, Style Variable, Text Flow & Justification

Page Body

The main body of a page.

Properties of Page Body

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Break Text

The text associated with a page break.

Properties of Page Break Text

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Margin, Padding, Relative Alignment, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible

Page Footer

The footer of a page.

Properties of Page Footer

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Header

The header of a page.

Properties of Page Header

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Number

The number of a page.

Properties of Page Number

Background Color, Classes, Font, Foreground Color, Margin, Number Style, Relative Alignment

Page Set

The set of pages to render according to a grouping structure.

Properties of Page Set

Grouping & Sorting, Master Detail Relationships, Name, Pagination, Properties, Query

Pareto Chart

A chart in which data series appear as colored sections stacked in columns or bars. The maximum of each column or bar represents the series total as a percentage of the overall total of all data series in the chart.

Negative values are not supported in pareto charts.

Properties of Pareto Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Chart Orientation, Classes, Conditional Palette, Conditional Styles, Connecting Lines, Cumulative Line, Depth, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Rotate Values, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible, Visual Angle

Pie Chart

A chart that uses sections of a circle as data markers to plot one or more data series. The size of each section is proportional to the value of each data series for a given category value. Each pie corresponds to a category value.

Properties of Pie Chart

Avoid Label Collision, Axis Title, Background Color, Background Image, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Depth, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Hole Size (%), Labels, Legend, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Pie Labels, Query, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Representation, Values, Visible

Pie Labels

The labels that will be drawn if multiple pie charts are rendered. If this object does not exist, no labels will be rendered.

Properties of Pie Labels

Classes, Conditional Styles, Drill-Through Definitions, Font, Foreground Color, Horizontal Alignment, Maximum Truncation Characters, Style Variable, Truncation, Truncation Text, Visible

Point Color Legend Title

The title for the point color legend. If this object is empty, a default title is rendered, if available.

Properties of Point Color Legend Title

Master Detail Relationships, Properties, Query

Point Layer

A map layer that includes points, such as cities. The color and size of each point is determined by their respective measure.

Properties of Point Layer

Border Color, Borders, Color Legend Title, Conditional Palette, Conditional Styles, Fill Effects, Labels, Map Drills, Palette, Show Data Range in Legend, Show Features with No Data, Size Legend Title, Style Variable, Values

Point Measure

The measure that determines the colors of points on a map chart.

Properties of Point Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Point Size Legend Title

The title for the point size legend. If this object is empty, a default title is rendered, if available.

Properties of Point Size Legend Title

Master Detail Relationships, Properties, Query

Point Size Measure

The measure that determines the size of the points on a map chart.

Properties of Point Size Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Maximum Size (pt), Member, Minimum Size (pt), Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Polar Chart

A point chart that plots one or more data series against two measures. The radius of a data series is determined by one measure and the arc is determined by the other measure. Multiple points are plotted for each category. They can be distinguished with the help of tool tips if the Tool Tip property is set to Yes.

Properties of Polar Chart

Angular Axis, Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Radial Axis, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Spider Effects, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Progressive Chart

A chart that uses columns as data markers to plot one category across a single measure. The top of the first column represents the starting value for the second column. This chart emphasizes the positive or negative contribution of each value to the total.

Properties of Progressive Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Chart Orientation, Classes, Conditional Palette, Conditional Styles, Connecting Lines, Depth, Drill-Through Definitions, Fill Effects, First Column Color, Floating, Font, Footer, Foreground Color, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, Negative Column Color, No Data Contents, Notes, Ordinal Axis, Padding, Pagination, Palette, Positive Column Color, Progressive Axis, Query, Relative Alignment, Render Page when Empty, Render Variable, Rotate Values, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Total Column, Value Location, Values, Visible, Visual Angle

Prompt Button

A predefined button used in prompt pages. Its usage changes according to its Type property, which can be set to Cancel, Back, Next, Finish, or Reprompt.

Properties of Prompt Button

Background Color, Background Image, Border, Classes, Conditional Styles, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Size & Overflow, Style Variable, Type

Query

The data that is to be retrieved from the database. The query consists of a source, a selection, detailed and summary filters, and dimension information.

Properties of Query

Auto Group & Summarize, Auto-Sort, Avoid Division by Zero, Cross Product Allowed, Define Member Sets, Execution Method, Execution Optimization, Generated SQL/MDX, Maximum Execution Time, Maximum Rows Retrieved, Maximum Tables, Maximum Text Blob Characters, Name, Outer Join Allowed, Override Dimension Info, Processing, Rollup Processing, SQL join syntax, Suppress, Use 1.x Behavior, Use for Parameter Info, Use Local Cache, Use SQL With Clause

Query Operation

Union, Intersect, Except (minus) operations on one or more queries that result in a projection list upon which other queries can be based.

Properties of Query Operation

Duplicates, Name, Projection List, Set Operation

Query Reference

A reference to another query defined in the same query set.

Properties of Query Reference

Cardinality

Radar Chart

A chart that integrates multiple axes into a single radial figure as lines or stacked areas.

Properties of Radar Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Ordinal Axis, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Radar Type, Radial Axis, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Spider Effects, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Radial Axis

The radial numeric axis for a polar chart or radar chart, including labels, titles, range, and scale.

Properties of Radial Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Radial Measure

The measure that determines the distance between the center of the chart and each data marker.

Properties of Radial Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Region Color Legend Title

The title for the region color legend. If this object is empty, a default title is rendered, if available.

Properties of Region Color Legend Title

Master Detail Relationships, Properties, Query

Region Layer

A map layer that includes regions, such as provinces.

Properties of Region Layer

Border Color, Borders, Color Legend Title, Conditional Palette, Conditional Styles, Fill Effects, Labels, Map Drills, Palette, Show Data Range in Legend, Show Features with No Data, Style Variable, Values

Region Measure

The measure that determines the colors of regions on a map chart.

Properties of Region Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Regression Line

A regression line for a bubble chart or scatter chart.

Properties of Regression Line

Box Type, Line Styles, Number of Regression Lines, Polynomial Exponent, Properties, Regression Type

Repeater

A table into which you can insert items that will be repeated.

Properties of Repeater

Grouping & Sorting, Master Detail Relationships, Name, No Data Contents, Pagination, Properties, Query, Render Page when Empty, Render Variable, Rows Per Page

Repeater Table

Renders query data in a table.

Properties of Repeater Table

Across, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Down, Floating, Font, Foreground Color, Grouping & Sorting, Horizontal Alignment, Margin, Master Detail Relationships, Name, No Data Contents, Pagination, Properties, Query, Relative Alignment, Render Page when Empty, Render Variable, Repeater Direction, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Repeater Table Cell

The contents of a repeater table object.

Properties of Repeater Table Cell

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Rich Text Item

Inserts an object that is used to render a subset of HTML in the layout. The HTML may come from either a static or dynamic source, and the object will also render in PDF output. For information about what elements are allowed in rich text items, see Inserting Other Objects.

Properties of Rich Text Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, Hierarchy, HTML, HTML Source Variable, Label, Level, Member, Name, Name, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Type

Row Number

Shows the row numbers in a column.

Properties of Row Number

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

Scatter Chart

A point chart that plots one or more data series against two measures. Multiple points are plotted for each category.

Properties of Scatter Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Conditional Palette, Conditional Styles, Drill-Through Definitions, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Regression Line, Relative Alignment, Render Page when Empty, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Select & Search Prompt

An advanced prompt control that allows you to search for values. You cannot use this prompt control with SAP BW data sources.

Properties of Select & Search Prompt

Box Type, Cascade Source, Case Insensitive, Conditional Styles, Data Format, Display Value, Floating, Hide Adornments, Multi-Select, Name, Parameter, Properties, Query, Render Variable, Required, Rows Per Page, Sorting, Static Choices, Style Variable, Use Value, Visible

Set Expression

A set of members.

Properties of Set Expression

Dimension, Expression, Hierarchy, Label, Name, Set Sorting, Type

Singleton

A query item that you can insert anywhere in the layout of your report where there is no query associated. When the report is run, the singleton object retrieves only the first row value for that query. Singletons are useful for adding boilerplate text to a report, such as a company name or address, to add overall calculations in a report, or to add multidimensional cell calculations in a report.

Properties of Singleton

Name, Properties, Query, Render Variable

Slicer Member Set

A set expression that returns members from a single hierarchy of a single dimension.

Properties of Slicer Member Set

Expression

SQL

An SQL query against a relational data source.

Properties of SQL

Data Source, Name, SQL, SQL Syntax

Summary Filter

A set of conditions in a query that narrow the scope of the data returned. A summary filter is applied after aggregation is complete.

Properties of Summary Filter

Expression, Scope, Usage

Table

A collection of cells in which objects can be organized in a grid fashion.

Properties of Table

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Pagination, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Table Cell

The cells within a row.

Properties of Table Cell

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Table of Contents

A collection of Table of Contents Entry objects that refer the same table of contents. The table of contents is generated in the rendered output.

Properties of Table of Contents

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, No Data Contents, Padding, Render Page when Empty, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Table of Contents Name, Text Flow & Justification, Visible, White Space

Table of Contents Entry

An entry in the table of contents.

Properties of Table of Contents Entry

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Heading Level, Horizontal Alignment, Margin, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Table of Contents, Text Flow & Justification, Visible, White Space

Table Row

The rows in a table.

Properties of Table Row

Background Color, Background Image, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Style Variable, Vertical Alignment

Target Measure

The target measure for a metrics chart.

Properties of Target Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Text Box Prompt

A prompt control that allows you to type in a value.

Properties of Text Box Prompt

Background Color, Border, Box Type, Conditional Styles, Default Selections, Floating, Font, Foreground Color, Hide Adornments, Hide Text, Multi-Line, Multi-Select, Name, Numbers Only,

Parameter, Range, Relative Alignment, Render Variable, Required, Size & Overflow, Style Variable, Use Thousands Separator, Visible

Text Item

A text item in a report. The content can be static text, or it can come from a query item or report expression.

Properties of Text Item

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Value, Drill-Through Definitions, Expression, Floating, Font, Foreground Color, Hierarchy, Label, Level, Margin, Maximum Characters, Member, Name, Name, Padding, Relative Alignment, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, Use Detail Value on Page, Visible

Time

The time when the report runs.

Properties of Time

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

Time Prompt

An advanced prompt control that allows you to select a time value.

Properties of Time Prompt

Box Type, Clock Mode, Conditional Styles, Default Selections, Display Milliseconds, Display Seconds, Floating, Hide Adornments, Multi-Select, Name, Parameter, Range, Render Variable, Required, Select UI, Style Variable, Visible

Tolerance Measure

The tolerance measure for a metrics chart.

Properties of Tolerance Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Total Column

A column or bar representing the total cumulative value of all other columns or bars in a progressive chart.

Properties of Total Column

Aggregate Function, Data Format, Data Item Value, Expression, Hierarchy, Label, Level, Member, Name, Properties, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Text, Total Column Color, Type

Total Column Label

The label to be rendered for the total column.

Properties of Total Column Label

Properties

Tree Prompt

A data driven prompt control that shows hierarchical information and allows you to select one or more members.

Properties of Tree Prompt

Cascade Source, Conditional Styles, Default Selections, Floating, Hide Adornments, Multi-Select, Name, Parameter, Pre-populate, Pre-populate Levels, Query, Render Variable, Required, Rows Per Page, Size & Overflow, Sorting, Style Variable, Use Value, Visible

Value Prompt

A prompt control that allows you to select one or more values from a list.

Properties of Value Prompt

Auto-Submit, Background Color, Box Type, Cascade Source, Conditional Styles, Data Format, Default Selections, Display Value, Floating, Font, Foreground Color, Hide Adornments, Multi-Select, Name, Parameter, Pre-populate, Properties, Query, Range, Relative Alignment, Render Variable, Required, Rows Per Page, Select UI, Size & Overflow, Sorting, Static Choices, Style Variable, Use Value, Visible

Variable

A report variable.

Properties of Variable

Name, Report Expression, Type

X Axis

The horizontal numeric axis for the chart, including labels, titles, range, and scale.

Properties of X Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

X Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of X Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Display Frequency, Drill-Through Definitions, First Label Index, Font, Foreground Color, Gridlines, Horizontal Alignment, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

X Axis Measure

The measure for the horizontal axis of a scatter chart or bubble chart.

Properties of X Axis Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Y1 Axis

The numeric axis of a metrics chart, or the primary numeric axis of a combination chart.

Properties of Y1 Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y2 Axis

The secondary numeric axis of a combination chart.

Properties of Y2 Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y Axis

The vertical numeric axis for the chart, including labels, titles, range, and scale.

Properties of Y Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of Y Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Display Frequency, Drill-Through Definitions, First Label Index, Font, Foreground Color, Gridlines, Horizontal Alignment, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

Y Axis Measure

The measure for the vertical axis of a scatter chart or bubble chart.

Properties of Y Axis Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Z Axis

The third numeric axis for a three-dimensional scatter chart, including labels, titles, range, and scale.

Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Z Axis

The vertical numeric axis for a three-dimensional combination chart, including labels, titles, range, and scale.

Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Z Axis Measure

The measure for the third axis of a three-dimensional scatter chart.

Properties of Z Axis Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Label, Level, Member, Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

Report Studio Properties

The following is a list of properties available in the lower left pane of Report Studio.

3-D Viewing Angle

Specifies the 3-D viewing angle of the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart

Absolute Position

Specifies that the legend will be positioned absolutely, by setting its pixel position.

Applies to

Legend

Across

Sets the number of times across, or horizontally, that the contents of the object may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is one. If it is set to Top to bottom, left to right, the default is 20.

Applies to

Repeater Table

Aggregate Function

Specifies the type of aggregation to apply. The Automatic setting means that the application groups or summarizes based on the data type. The Summarize setting means that any setting found in the model will be used to determine the type of aggregation. For more information about each summary function, see the Report Studio Professional Authoring User Guide.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Allow 45° Rotation

Specifies whether the labels can be rotated 45 degrees if the labels are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow 90° Rotation

Specifies whether the labels can be rotated 90 degrees if the labels are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow Skip

Specifies whether some labels can be skipped if they are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow Stagger

Specifies whether the labels can be staggered if they are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Angular Axis

Specifies whether the axis is rendered.

Applies to

Polar Chart

Application

Specifies if the condition will be applied before or after aggregation and summarization. When true, the condition will apply to the summarized rows, and a fact, or aggregate, in the expression will be interpreted as the aggregated value of the summarized rows. When false, the condition will apply to the detail database rows from the tabular result set prior to aggregation or summarization, and a fact, or aggregate, in the expression will be interpreted as the individual database value before it has been summarized. This property has no effect on OLAP data sources, on references to non-aggregate items, or when automatic summarization is disabled.

Applies to

Detail Filter

Apply Single Class

Specifies whether to apply all of the classes defined or only one class. When set to Yes, the last class that was applied is used. For example, if you applied a class to a crosstab intersection and another class to the member fact cells of a crosstab row, the class applied to the intersection is the last class applied to the fact cells. For information about the order in which classes are applied, see the Report Studio Professional Authoring Guide.

This property is set to Yes when you are upgrading a report so that the upgraded report will look the same as the original report. This is because objects in reports created using earlier versions of Cognos 8 support only one class.

Applies to

Crosstab Fact Cells

Auto Group & Summarize

Specifies whether the application will apply suggested aggregate functions to aggregate data items and group all non-aggregate data items, producing groups and summary rows. If it is set to No, detail rows will be rendered.

Applies to

Query

Auto-Sort

When running the report, specifies whether to automatically sort based on data type.

Applies to

Query

Auto-Submit

Specifies whether the application submits the prompt page automatically, as soon as a value is changed.

Applies to

Value Prompt

Auto Truncation

Specifies whether to allow truncation of text.

Applies to

Legend

Avoid Division by Zero

Specifies whether the application will return a null value when it encounters a division by zero. This property applies only to relational data sources.

Applies to

Query

Avoid Label Collision

Controls how labels are arranged. If set to false, the chart uses the default positions. If set to true, the chart uses a different layout to avoid label collision. To keep existing reports unchanged, set this property to false.

Applies to

Pie Chart

Axis Assignment

Specifies the numeric axis to use.

Applies to

Area, Bar, Baseline, Line, Marker

Axis Assignment

Specifies which numeric axis to use.

Applies to

Baseline

Axis Labels

Specifies whether to show or hide axis labels.

Applies to

Angular Axis, Cumulation Line Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Axis Line

Specifies the properties of the axis line in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Axis Title

Specifies whether an axis title will be rendered.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Chart, Map, Numerical Axis, Numerical Axis, Ordinal Axis, Pie Chart, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Background Color

Specifies the background color for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Chart Body, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Date, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Legend, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Page, Page Body, Page Break Text, Page Footer, Page Header, Page Number, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart,

Repeater Table, Repeater Table Cell, Row Number, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Box Prompt, Text Item, Time, Value Prompt

Background Image

Specifies an image to be used as the background for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Chart Body, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Legend, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Item

Bar, Line, or Area Index

Specifies which combination object to use when calculating the position. When multiple series exist, the index of 0 represents the topmost bar, line, or area in the Series drop zone, the index of 1 represents the second one, and so on.

Applies to

Baseline, Baseline, Marker

Baselines

Adds reference lines to a chart based on numeric or statistical values, calculations, or layout calculations.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Block Variable

Specifies a variable based on which the block can be conditionally rendered.

Applies to

Conditional Block, Conditional Block

Border

Specifies the width, style, and color for the border of the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Text Box Prompt, Text Item

Border Color

Specifies the color of the border rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

Applies to

3-D Area, 3-D Bar, 3-D Line, 3-D Scatter Chart, Area, Bar, Bubble Chart, Display Layer, Gauge Chart, Legend, Line, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Borders

Specifies whether borders are rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

Applies to

3-D Area, 3-D Bar, 3-D Line, 3-D Scatter Chart, Area, Bar, Bubble Chart, Display Layer, Gauge Chart, Legend, Line, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Bottom Position (px)

Specifies the bottom position of the note, in pixels.

Applies to

Note

Bottom Position (px)

Specifies the pixel position of the bottom edge of the legend.

Applies to

Legend

Box Type

Specifies whether to override the default box type for the object. When set to None, the object is not rendered and its space is not reserved in the report. When set to Inline, you can insert other objects on the same line as the object. When set to Block, you can insert other objects only on the lines above and below the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Field Set, Gauge Chart, Hyperlink, Image, List, Map, Metrics Range Chart, Page Break Text, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater Table, Scatter Chart, Table, Table of Contents, Table of Contents Entry, Text Item

Box Type

Specifies whether to override the default box type for the object.

Applies to

Axis Title, Baseline, Baseline, Baseline, Chart Footer, Chart Subtitle, Chart Title, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Date & Time Prompt, Date Prompt, Interval Prompt, Legend, Legend Title, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Marker, Marker, Note Content, Page Body, Page Footer, Page Header, Regression Line, Repeater Table Cell, Select & Search Prompt, Table Cell, Table Row, Text Box Prompt, Time Prompt, Value Prompt

Calculation Intersection

Specifies that if an edge member intersects with a calculated member, the resulting cell value should be N/A.

Applies to

Calculated Measure, Calculated Member, Data Item

Calendar Type

Specifies the type of calendar to show. The date values are mapped to the selected calendar before being formatted. The default value is inherited from the user's content language.

Applies to

Date & Time Prompt, Date Prompt

Caption

Specifies the caption for the level.

Applies to

Level

Caption

Specifies the caption.

Applies to

Calculated Member

Cardinality

Specifies the cardinality for this join operand.

Applies to

Query Reference

Cascade Source

Specifies the parameter whose value is used to filter the values displayed in this control.

Applies to

Select & Search Prompt, Tree Prompt, Value Prompt

Case Insensitive

Specifies whether to perform a case insensitive search by default.

Applies to

Select & Search Prompt

Catalog

Specifies the OLAP catalog.

Applies to

MDX

Chart Orientation

Specifies whether the chart is rendered vertically or horizontally.

Applies to

Combination Chart, Pareto Chart, Progressive Chart

Chart Type

Specifies whether the data may be rendered as either a bar, line, or area.

Applies to

Area, Bar, Line

Chart Type

Specifies whether the data may be rendered as either a bar, line or area.

Applies to

3-D Area, 3-D Bar, 3-D Line

Classes

Specifies a class to apply to the object. The class provides a default style. If you apply more than one class, the style properties from all classes are merged together when they are applied. However, if the classes have style properties in common, the style properties from the last class applied override those from previous classes.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Axis, Date, Field Set, Gauge Chart, Gauge Labels, Hyperlink, Hyperlink Button, Image, Legend, Legend Title, List, List Cell, List Column Body, List Column Title, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Break Text, Page Footer, Page Header, Page Number, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Row Number,

Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Item, Time, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Clock Mode

Specifies whether the arms of the clock move.

Applies to

Date & Time Prompt, Time Prompt

Color Legend Title

Specifies a title within the legend above the palette for the region color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to

Region Layer

Color Legend Title

Specifies a title within the legend above the palette for the point color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to

Point Layer

Column Coordinate

Uniquely identifies the column of a node member or spacer on an edge of the crosstab. You cannot modify this value.

Applies to

Crosstab Intersection

Column Titles

Specifies where or whether column titles may be rendered.

Applies to

List

Component Reference

Specifies the layout object that is referenced. An object is a reusable component only if it has a name.

Applies to

Layout Component Reference

Component Reference

Specifies the layout object that is referenced. An object is a reusable component only if it has a name. You cannot modify this value.

Applies to

Component Override

Conditional Palette

Specifies a conditional palette for the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Conditional Styles

Specifies the conditions and styles used to style the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Angular Measure, Axis Title, Block, Bubble Chart, Bubble Measure, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Text Item, Chart Title, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date & Time Prompt, Date Prompt, Default Measure, Display Layer, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Hyperlink Button, Image, Interval Prompt, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Point Layer, Point Measure, Point Size Measure, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Radial Measure, Region Layer, Region Measure, Repeater Table, Repeater Table Cell, Scatter Chart, Select & Search Prompt, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Target Measure,

Text Box Prompt, Text Item, Time Prompt, Tolerance Measure, Tree Prompt, Value Prompt, X Axis, X Axis, X Axis Measure, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Y Axis Measure, Z Axis, Z Axis, Z Axis Measure

Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar. This property is ignored for clustered bars.

Applies to

Bar, Pareto Chart, Progressive Chart

Contents Height

Specifies the relative height of list rows. This property is used only when a list has a height defined in the Size and Overflow property.

Stretched means that the rows will be evenly sized to fit in the list's height. This is default HTML behavior.

Minimal means that rows will take up only as much space as they need, and be compressed at the top of the list. You can position a footer at the bottom of the list by setting the Push To Bottom property to Yes on a footer object inside the list.

Applies to

List

Cross Product Allowed

Specifies whether the query will be allowed to run if there is a cross join between database tables. This type of query generates a result set that includes all possible unique combinations of values from the first and second table. The default value is Deny.

Applies to

Query

Cumulation Axis

Specifies whether the cumulation axis may be rendered.

Applies to

Cumulation Line

Cumulation Label

Specifies whether a label for the cumulation line may be rendered.

Applies to

Cumulation Line

Cumulative Line

Specifies whether the cumulation line is rendered.

Applies to

Pareto Chart

Current Block

Specifies which block is currently being authored.

Applies to

Conditional Block, Conditional Block

Custom Label

Overrides the default label for the data item.

Applies to

Angular Measure, Bubble Measure, Chart Node Member, Default Measure, Point Measure, Point Size Measure, Radial Measure, Region Measure, Target Measure, Tolerance Measure, X Axis Measure, Y Axis Measure, Z Axis Measure

Data Format

Specifies the data format of the object.

Applies to

Angular Axis, Angular Measure, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date, Default Measure, Gauge Numerical Axis, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Numerical Axis, Numerical Axis, Point Measure, Point Size Measure, Radial Axis, Radial Measure, Region Measure, Row Number, Select & Search Prompt, Target Measure, Text Item, Time, Tolerance Measure, Total Column, Value Prompt, X Axis, X Axis Measure, Y1 Axis, Y2 Axis, Y Axis, Y Axis Measure, Z Axis, Z Axis, Z Axis Measure

Data Item

Specifies a reference to a data item. You cannot modify this value.

Applies to

Crosstab Node Member, Member Set

Data Item

Specifies a reference to a data item.

Applies to

Calculated Member, Fact, Key, Member Property

Data Item Label

Specifies the data item label that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Data Item Label

Specifies the data item label that defines the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Data Item Label

Specifies the data item label that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Data Item Label

Specifies the data item label that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Data Item Value

Specifies the data item value that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Data Item Value

Specifies the data item value that defines the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Data Item Value

Specifies the data item value that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Data Item Value

Specifies the data item value that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Data Item Value

Specifies the numeric position by using a data item value.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Data Item Value

Specifies the numeric position of a data item.

Applies to

Marker

Data Language

Specifies the language of the data.

Applies to

Map

Data Source

Specifies the query data source.

Applies to

MDX, SQL

Default Measure

Specifies the default measure to use for a crosstab or chart. If the measures of the crosstab or chart cannot be determined by what is being rendered on the edges, then the default measure will be rendered.

Applies to

Crosstab

Default Selections

Specifies the collection of default selections for a prompt control.

Applies to

Date & Time Prompt, Date Prompt, Interval Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Default Title

Specifies whether the default title may be generated.

Applies to

Axis Title, Legend Title

Define Contents

Overrides the content of the selected crosstab intersection. Use this property to hide measure values for individual cells or to define custom content.

Applies to

Crosstab Intersection, Crosstab Member Fact Cells

Define Member Sets

Specifies the set structure of a query. If it is not defined, it is assumed that each data item defines an unrelated set.

Applies to

Query

Depth

Specifies the depth of the chart. A value of zero indicates a flat chart.

Applies to

Combination Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Progressive Chart

Description

Specifies a description for the object, that is used to assist authoring.

Applies to

Class, HTML Item, Metric Studio Diagram, Rich Text Item

Detail

Specifies whether the data item is to be used for calculating aggregates or not. When set to Yes, the data item is used to aggregate the lowest level details.

Applies to

Data Item

Diagram Identifier

Identifies a diagram in Metric Studio.

Applies to

Metric Studio Diagram

Dial Outline Color

Specifies the dial outline color in a gauge chart.

Applies to

Gauge Chart

Dictionary

Specifies the aliases to use when matching data values to feature names in the map.

Applies to

Map

Dimension

Specifies a reference to a dimension. You cannot modify this value.

Applies to

Calculated Measure, Calculated Member, Explicit Member Set, Hierarchy Set, Intersection (Tuple) , Level Set, Measure, Member, Member Children Set, Member Property, Named Set, Set Expression

Display After Overall Header

Specifies whether the list page header is to be rendered after the overall header.

Applies to

List Page Header

Display Frequency

Specifies the frequency for which chart labels are to be rendered. If set to 3, for example, every third label will be rendered.

Applies to

Ordinal Axis, X Axis, Y Axis

Display Milliseconds

Specifies whether to show the milliseconds. The format of the milliseconds can be controlled by selecting a specific format. This property is ignored if seconds are not rendered. The default value is inherited from the user's content language.

Applies to

Date & Time Prompt, Interval Prompt, Time Prompt

Display Seconds

Specifies whether to show the seconds. The format of the seconds can be controlled by selecting a specific format. The default value is inherited from the user's content language.

Applies to

Date & Time Prompt, Interval Prompt, Time Prompt

Display Value

Specifies the values rendered to the report user when the prompt is used. These values can be different than the ones that are actually used by the report.

Applies to

Select & Search Prompt, Value Prompt

Down

Specifies the number of times down, or rows, that the frame contents may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is 20. If it is set to Top to bottom, left to right, the default is one.

Applies to

Repeater Table

Drill-Through Definitions

Specifies report-to-report drill-through definitions, bookmarks, and drill-through text for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Gauge Chart, Gauge Labels, Image, Legend, List Column Body, List Column Title, Map, Metrics Range Chart, Ordinal Axis, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart, Text Item, X Axis, Y Axis

Duplicates

Specifies whether duplicate rows will be preserved or removed.

Applies to

Query Operation

Embed

Specifies how to embed the reference object. A reference, or pointer, to the object is stored, by default. Alternatively, a copy of the external object can be stored in the report.

Applies to

Layout Component Reference

Execution Method

Specifies whether the query is a candidate to run concurrently. If set to Concurrent, the query may still execute sequentially based on other factors. If not explicitly set, the query will execute sequentially. Concurrent query execution may improve performance in some cases.

Applies to

Query

Execution Optimization

Specifies how much of the query processing is performed by the client and how much is performed by the database server. If the database server can perform all the query processing, it does.

If All Rows is selected, the optimizer adopts a plan that retrieves all rows of the result set in the least amount of time. This value is generally used in a batch environment.

If First Rows is selected, the optimizer adopts a plan that retrieves the first row as quickly as possible. This value is generally used in an interactive environment.

If Incremental is selected, the optimizer retrieves the first N rows, and then retrieves the next N rows.

Applies to

Query

Expand Features

Specifies whether to center and expand the feature in the map chart. When set to Yes, the map feature is centered and expanded to take up all available space in the chart. When set to No, the map feature is not expanded.

Applies to

Map

Expression

Specifies the expression to evaluate when filtering the data.

Applies to

Detail Filter, Summary Filter

Expression

Specifies the expression that defines the slicer member set.

Applies to

Slicer Member Set

Expression

Specifies the numeric position for a data marker in a scatter chart.

Applies to

Marker

Expression

Specifies the expression used to populate the data item.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Calculated Measure, Calculated Member, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Face Color

Specifies the color to show on the face of each gauge in a gauge chart.

Applies to

Gauge Chart

Fact Cells Precedence

Specifies which style property will override the other style property for intersecting cells in a crosstab, the row's properties, or the column's properties.

This property only applies to style properties that are both set, but to different values. For example, if the row's background color is set to yellow and the column's background color is set to red, you can select which of these properties will override the other. If only the row or the column has a set background color, then that color will be used on the intersecting cell, regardless of this setting.

Applies to

Crosstab

Fill Effects

Specifies the fill effects for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Chart Body, Combination Chart, Display Layer, Gauge Chart, Legend, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

First Column Color

Specifies the color, gradient, or pattern to be used for the first column in the progressive chart.

Applies to

Progressive Chart

First Date

Specifies the earliest date to render in the control, and the earliest date that can be selected.

Applies to

Date & Time Prompt, Date Prompt

First Label Index

Specifies which label will be rendered first. If set to 5, for example, the fifth label will be the first label rendered. Subsequent labels will be rendered as defined in the Display Frequency property.

Applies to

Ordinal Axis, X Axis, Y Axis

Floating

Specifies how objects flow around an object.

Float controls the way the content that follows the selected object will flow around or below it.

Clear controls where the selected object is positioned, relative to other floating objects.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, Map, Metrics Range Chart, Page Break Text, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Scatter Chart, Select & Search Prompt, Table, Table of Contents, Table of Contents Entry, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Font

Specifies the font family, size, weight, style, and effects used to display the object's text.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Break Text, Page Footer, Page Header, Page Number, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Row Number, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Box Prompt, Text Item, Time, Value Prompt, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Font Auto-Sizing

Specifies whether to allow automatic resizing of the font.

Applies to

Legend

Footer

Specifies whether a chart footer is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Foreground Color

Specifies the color of the object's text.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Hyperlink Button, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Break Text, Page Footer, Page Header, Page Number, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Row Number, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Box Prompt, Text Item, Time, Value Prompt, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Gauge Labels

Specifies whether gauge labels are rendered.

Applies to

Gauge Chart

Gauge Palette

Specifies the palette that controls the look of the dial portion of a gauge.

Applies to

Gauge Chart

Generated SQL/MDX

Shows the generated SQL or MDX.

Applies to

Query

Gridlines

Specifies the properties of the gridlines in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Grouping & Sorting

Specifies the grouping and sorting structure.

Applies to

List, Page Set, Repeater, Repeater Table

Grouping Type

Specifies whether the absolute, stacked, or 100 percent stacked data may be drawn.

Applies to

Area, Bar, Line

Group Span

Specifies the group that this cell should visually span.

Applies to

List Column Body

Has Fact Cells

Specifies the contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

Applies to

Crosstab

Heading Level

Specifies the heading level of the table of contents entry.

Applies to

Table of Contents Entry

Height (px)

Specifies the height of the note, in pixels.

Applies to

Note

Hide Adornments

Specifies whether to hide the asterisk (*) on required prompts and arrow (->) on type-in prompts that are in an error state.

Applies to

Date & Time Prompt, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Hide Text

Specifies whether to replace characters entered in the prompt control with asterisk (*) characters.

Applies to

Text Box Prompt

Hierarchy

Specifies a reference to a hierarchy. You cannot modify this value.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Calculated Member, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, Explicit Member Set, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Member, Member Children Set, Member Property, Named Set, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Hole Size (%)

Specifies the size of the hole in a donut chart. A value of zero indicates a pie chart.

Applies to

Pie Chart

Horizontal Alignment

Specifies how the contents of the selected object is aligned horizontally.

Applies to

Block, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Labels, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Note Content, Ordinal Axis, Page, Page Body, Page Footer, Page Header, Pie Labels, Repeater Table, Repeater Table Cell, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, X Axis, Y Axis

HTML

Specifies the static text used as HTML.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

HTML Source Variable

Specifies a variable based on which the HTML source is chosen.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Ignore Data with No Features

Specifies whether to allow data that has no corresponding features. When set to Yes, data with no corresponding features will be ignored. When set to No, the map will not run if it contains data with no corresponding features.

Applies to

Map

Include Zero For Auto Scale

Specifies whether the value zero is included in the automatic calculation of the numeric scale. It is only relevant if the chart is involved in a master detail relationship.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

Join Relationships

Specifies how to join the two queries.

Applies to

Join

Label

Specifies the static text that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Label

Specifies the class label for a local class.

Applies to

Class

Label

Specifies the class label for a global class. You cannot modify this label.

Applies to

Class

Label

Specifies whether a label is rendered for the baseline.

Applies to

Baseline, Baseline

Label

Specifies whether a label is rendered for the marker.

Applies to

Marker, Marker

Label

Specifies the label of the object.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Calculated Measure, Calculated Member, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Explicit Member Set, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Measure, Member, Member Children Set, Member Property, Named Set, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Label Control

Controls how the labels in a chart are rendered.

Applies to

Ordinal Axis, X Axis, Y Axis

Labels

Specifies whether labels are rendered in the chart.

Applies to

Display Layer, Point Layer, Region Layer

Labels

Specifies whether labels are rendered.

Applies to

Pie Chart

Last Date

Specifies the latest date rendered in the control, and the last date that can be selected.

Applies to

Date & Time Prompt, Date Prompt

Left Position (px)

Specifies the left position of the note, in pixels.

The position is measured from the left edge of the note.

Applies to

Note

Left Position (px)

Specifies the pixel position of the left edge of the legend.

Applies to

Legend

Legend

Specifies whether the legend is rendered.

Applies to

Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Radar Chart, Scatter Chart

Legend Label

Specifies whether to render the baseline in the legend.

Applies to

Baseline, Baseline

Legend Title

Specifies whether a legend title is rendered.

Applies to

Legend

Level

Specifies a reference to a level. You cannot modify this value.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, HTML Item, Image, Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Member Property, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Level Indentation

Controls the indentation of the contents of node members in a crosstab.

Applies to

Crosstab Node Member

Line

Specifies whether a line may be rendered. This allows you to show markers without lines.

Applies to

Line

Line Style

Specifies the style of the line.

Applies to

Line

Line Styles

Specifies the line style, color, and weight of the line.

Applies to

Baseline, Baseline, Baseline, Baseline, Cumulation Line, Regression Line

Line Weight (pt)

Specifies the line thickness in points. A value of zero indicates the thinnest possible line.

Applies to

Line

Map & Layers

Sets the map and layers for a map chart.

Applies to

Map

Map Drills

Controls the drill definitions in a map. It is possible to define different drill targets for each region or point.

Applies to

Point Layer, Region Layer

Margin

Specifies the margin properties for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Date, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, Map, Metrics Range Chart, Page, Page Break Text, Page Number, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Row Number, Scatter Chart, Table, Table of Contents, Table of Contents Entry, Text Item, Time

Marker Color

Specifies a color for the target value markers in a metrics chart.

Applies to

Metrics Range Chart

Marker Label

Specifies whether the label for the Target Marker will appear in the legend.

Applies to

Metrics Range Chart

Markers

Adds reference points to a chart based on numeric or statistical values, calculations, or layout calculations.

Applies to

Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Marker Text Location

Specifies where the text of the marker is rendered.

Applies to

Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Master Detail Relationships

Specifies relationships between the master data container and the detail data container. Specifically, specifies how query items in the master query are linked to query items or parameters in the detail query.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Axis Title, Baseline, Baseline, Baseline, Baseline, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Crosstab, Gauge Chart, Legend Title, List, Map, Marker, Marker, Metrics Range Chart, Note Content, Page, Page Set, Pareto Chart, Pie Chart, Point Color Legend Title, Point Size Legend Title, Polar Chart, Progressive Chart, Radar Chart, Region Color Legend Title, Repeater, Repeater Table, Scatter Chart

Maximum Characters

Specifies the maximum number of characters to show before the text is truncated.

Applies to

Legend, Text Item

Maximum Execution Time

Specifies the maximum period, in seconds, that the query can spend to open the database cursor and to retrieve the first row of data. An error is returned if the specified time is exceeded. Note that

this property is not for the total time required to execute the query. If no value is specified, no error is returned and the query runs until complete.

Applies to

Query

Maximum Hotspots

Specifies the maximum number of hotspots generated in a chart. If specified, this value overrides the hotspot configuration settings in Cognos Administration.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Maximum Rows Retrieved

Specifies the maximum number of database rows that the query can retrieve. An error is returned if the number of database rows returned exceeds the specified value. If no value is specified, no error is returned and the query returns all rows.

Applies to

Query

Maximum Size (pt)

Specifies the maximum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.

Applies to

Point Size Measure

Maximum Tables

Specifies the maximum number of tables that the query can retrieve. An error is returned if the number of tables in the generated Cognos SQL exceeds the specified value. If no value is specified, no error is returned and there is no restriction on the number of tables that can be queried.

Applies to

Query

Maximum Text Blob Characters

Specifies the maximum number of characters that the query is allowed to retrieve for each text BLOB. An error is returned if the number of characters retrieved exceeds the specified value. If no value is specified, no error is returned and text BLOBs can be of any size.

Applies to

Query

Maximum Truncation Characters

Specifies the maximum number of characters to show before the label is truncated.

If the Allow Truncation property is set to Yes and no value is specified in this property, the application will automatically determine the optimum number of characters after which to truncate. Use this property only if you want explicit control over the truncation level. Note that regardless of this property's setting, no truncation will occur if there is sufficient space.

Applies to

Gauge Labels, Ordinal Axis, Pie Labels, X Axis, Y Axis

Maximum Value

Specifies the maximum value for the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

MDX

The text of the typed-in MDX. It is assumed to be appropriate for the type and data source. If it is not the query may fail or produce unexpected results.

Applies to

MDX

Member

Specifies a reference to a member. You cannot modify this value.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, HTML Item, Image,

List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Measure, Member, Member Children Set, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Member Offset (%)

Specifies a position relative to the next item in the chart. This is a percentage value. Negative values indicate before the member and positive values indicate after the member.

Applies to

Baseline

Members

Specifies the members of the intersection (tuple).

Applies to

Intersection (Tuple)

Minimum Size (pt)

Specifies the minimum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.

Applies to

Point Size Measure

Minimum Value

Specifies the minimum value for the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

Minor Gridlines

Specifies the properties of the minor gridlines in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Multi-Line

Specifies whether to allow multi-line editing in the text control

Applies to

Text Box Prompt

Multi-Select

Specifies whether the control allows the selection of multiple values. Note that an associated parameterized filter expression will override this object's setting. If you edit this property but do not get the expected results, check the associated expression for the presence of an operator that specifies how many items can be selected. Examples of multiple selection operators are "in" and "not in"; examples of single selection operators are equal (=), less than (<) and greater than (>).

For example, if you used the prompt wizard to create a parameterized filter expression and selected one of the operators that specify selection rules, changing the value of this property is not sufficient to change this restriction. You must also edit the filter's expression to remove the offending operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

Applies to

Date & Time Prompt, Date Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Name

Specifies the name of the object.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Calculated Measure, Calculated Member, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Dimension, Explicit Member Set, Fact, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level, Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Measure, Member, Member Children Set, Member Property, Member Set, Named Set, Page, Page Set, Point Measure, Point Size Measure, Query, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, Variable, X Axis Measure, Y Axis Measure, Z Axis Measure

Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied styling.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Combination Chart, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Generated Prompt, HTML Item, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater, Repeater Table, Rich Text Item, Scatter Chart, Select & Search Prompt, Singleton, Table, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Name

Specifies the unique name identifier for a query operation.

Applies to

Query Operation

Name

Specifies the unique name identifier for an SQL object.

Applies to

SQL

Name

Specifies the unique name identifier for an MDX object.

Applies to

MDX

Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied formatting.

Applies to

Calculated Member, Key, Level Hierarchy, Member Hierarchy, Member Property

Negative Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent negative values.

Applies to

Progressive Chart

No Data Contents

Specifies whether to show the no data contents tab for the selected query frame. When set to Yes, you can specify on this tab what to show when there is no data. When set to No, the tab is hidden and the query frame reverts to the default behavior.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab, Gauge Chart, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Table of Contents

No Data Features Size (pt)

Specifies the point size used for map point features that do not have corresponding data.

Applies to

Map

Node Coordinate

Uniquely identifies a node member or spacer on an edge of a crosstab. Used by the Crosstab Intersection object to uniquely identify the intersection of elements from each edge. You cannot modify this value.

Applies to

Crosstab Node Member, Crosstab Space

Note Border

Specifies the properties for the border of a note.

Applies to

Note

Notes

Specifies a block of text that you can position on a chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Number of Regression Lines

Specifies whether there will be one regression line for all the data or one for each series.

Applies to

Regression Line

Numbers Only

Specifies whether the Text Box Prompt allows numbers only.

Applies to

Text Box Prompt

Number Style

Specifies the style to use for page numbers.

Applies to

Page Number

Numerical Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

Numerical Axis

Specifies whether the numeric axis of a gauge chart is rendered.

Applies to

Gauge Chart

Numeric Value

Specifies the numeric position by using a number.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Numeric Value

Specifies the value of the numeric position.

Applies to

Marker

Ordinal Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart, Metrics Range Chart, Progressive Chart, Radar Chart

Outer Join Allowed

Specifies whether outer joins are allowed on the object. This property applies to a single query and overrides the setting in Framework Manager, the modeling tool.

Applies to

Query

Override Dimension Info

Specifies dimension information for a query.

Applies to

Query

Overrides

Specifies whether to override child objects. Before you can override child objects, they must have a name.

Applies to

Layout Component Reference

Padding

Specifies the space between the object and the margin. If there is a border then it specifies the space between the object and the border.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Gauge Chart, Hyperlink, Hyperlink Button, List Cell, List Column, List Column Body, List Columns, List Columns,

List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table Cell, Scatter Chart, Table Cell, Table of Contents, Table of Contents Entry, Text Item

Pagination

Specifies pagination rules, such as page breaks, keep-with properties, page counts, and numbering.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab, Crosstab Node Member, Crosstab Space, Gauge Chart, List, List Column, List Header, Map, Metrics Range Chart, Page, Page Set, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Table

Palette

Specifies the palette to use for the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Palette

Map layers use the numeric palette, in which the color of a region or point is based on its numeric value.

Applies to

Point Layer, Region Layer

Parameter

Specifies the parameter that is satisfied by values chosen in the prompt control.

Applies to

Date & Time Prompt, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

PDF Page Setup

Specifies the orientation and paper size for PDF documents.

Applies to

Page

Percentile

Specifies a position based on a data percentile value. This value must be greater than zero.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker, Marker

Percent of Axis

Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker, Marker

Performance Pattern

Controls what portions of the range markers for tolerance and target ranges are rendered on a metrics chart.

Applies to

Metrics Range Chart

Pie Labels

Specifies whether pie labels are rendered.

Applies to

Pie Chart

Point Color

Specifies the color of the markers.

Applies to

Marker, Marker

Point Shape

Specifies the shape of the markers.

Applies to

Cumulation Line, Marker, Marker

Point Shape

Specifies the shape of the markers. If you choose a value of series or category, the marker shape varies accordingly.

Applies to

3-D Scatter Chart, Bubble Chart, Line, Polar Chart, Radar Chart, Scatter Chart

Point Size (pt)

Specifies the size of markers in points. A value of zero means do not show markers.

Applies to

3-D Scatter Chart, Cumulation Line, Line, Marker, Marker, Polar Chart, Radar Chart, Scatter Chart

Polynomial Exponent

Specifies the highest exponential value to use in the regression calculation.

Applies to

Regression Line

Position

Specifies where to position the legend.

Applies to

Legend

Position type

Specifies the source type for the numeric position.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Position type

Specifies the type of numeric position.

Applies to

Marker

Position type

Specifies a type of numeric position.

Applies to

Marker

Positive Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent positive values.

Applies to

Progressive Chart

Pre-populate

Specifies whether to pre-populate the control with values, but only if the parent of this prompt control is optional. This only applies to prompt controls that have a parent in a cascade.

Applies to

Tree Prompt, Value Prompt

Pre-populate Levels

Specifies the number of levels to pre-populate the prompt with. The default value is 1, which will pre-populate the prompt with only the root members.

Applies to

Tree Prompt

Pre-Sort

Sorts the data that is used by the query to produce a temporary cube when needed, such as for a crosstab against a relational data source. This property affects the default order of members of a level populated from the data item.

Applies to

Data Item

Processing

Specifies whether the query engine will pick up a minimal amount of processing. Local processing only occurs if the database cannot handle the load.

Applies to

Query

Progressive Axis

Specifies whether the axis is rendered.

Applies to

Progressive Chart

Projection List

Shows the list of projected data items for the set operation. You can automatically generate the list or manually add data items.

Applies to

Query Operation

Properties

Specifies a list of data items from the query in scope to associate to the selected object. This is necessary when you want to reference a data item that is in the associated query, but not used in the layout. For more information, see the Report Studio Professional Authoring User Guide.

Applies to

Axis Title, Baseline, Baseline, Baseline, Baseline, Chart Footer, Chart Node Member, Chart Subtitle, Chart Title, Crosstab Node Member, Cumulation Line, Legend Title, List, List Footer, List Header, Marker, Marker, Note Content, Page, Page Set, Point Color Legend Title, Point Size Legend Title, Region Color Legend Title, Regression Line, Repeater, Repeater Table, Select & Search Prompt, Singleton, Total Column, Total Column Label, Value Prompt

Property

Specifies a reference to a member property. You cannot modify this value.

Applies to

Member Property

Push To Bottom

Specifies whether to position the footer as low as possible inside the parent object.

Applies to

List Footer, List Page Footer

Query

Specifies a reference to a query.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Axis Title, Baseline, Baseline, Baseline, Baseline, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Crosstab, Gauge Chart, Legend Title, List, Map, Marker, Marker, Metrics Range Chart, Note Content, Page, Page Set, Pareto Chart, Pie Chart, Point Color Legend Title, Point Size Legend Title, Polar Chart, Progressive Chart, Radar Chart, Region Color Legend Title, Repeater, Repeater Table, Scatter Chart, Select & Search Prompt, Singleton, Tree Prompt, Value Prompt

Radar Type

Specifies how the radar chart is rendered.

Applies to

Radar Chart

Radial Axis

Specifies whether the axis is rendered.

Applies to

Polar Chart, Radar Chart

Range

Specifies whether this control accepts ranges. The setting of the associated parameterized expression for this property will override the setting of this object. If you edit this property but do not get the expected results, check the associated expression for the presence or absence of an in_range operator.

For example, if you created this prompt control with the prompt wizard and set up an associated parameterized filter that accepts ranges, changing the value of this property is not sufficient to change this restriction. You must also edit the filter's expression and remove the in_range operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

Applies to

Date & Time Prompt, Date Prompt, Interval Prompt, Text Box Prompt, Time Prompt, Value Prompt

Range Label

Specifies whether the label for the Target Range will appear in the legend.

Applies to

Metrics Range Chart

Regression Line

Specifies whether a regression line is rendered.

Applies to

Bubble Chart, Scatter Chart

Regression Type

Specifies the type of regression used.

Applies to

Regression Line

Relative Alignment

Specifies how to vertically align this object, relative to its siblings.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Class, Combination Chart, Context Item Text, Crosstab, Date, Date Prompt, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, Map, Metrics Range Chart, Page Break Text, Page Number, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Row Number, Scatter Chart, Table, Text Box Prompt, Text Item, Time, Value Prompt

Render Fact Cells

Specifies whether to render values in the fact cells of the spacer on an edge of the crosstab.

Applies to

Crosstab Space

Render Page when Empty

Specifies whether to render a page when data containers on a page contain no data. If all data containers on a page have this property set to No and do not have any data to render, the page is not rendered. If any data container on a page has data or has this property set to Yes, the page is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab, Gauge Chart, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Table of Contents

Render Variable

Specifies a variable based on which the object can be conditionally rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Combination Chart, Context Item Text, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Generated Prompt, HTML Item, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, List Column, Map, Metrics Range Chart, Page, Page Break Text, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Repeater Table Cell, Rich Text Item, Scatter Chart, Select & Search Prompt, Singleton, Table, Table of Contents, Table of Contents Entry, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Repeater Direction

Specifies the direction in which to populate the rendered repeater cells.

Applies to

Repeater Table

Report Expression

Specifies the report expression that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Report Expression

Specifies the report expression used to define the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Report Expression

Specifies the report expression that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Report Expression

Specifies the report expression that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Report Expression

Specifies the numeric position by using a report expression.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Report Expression

Specifies the report expression for the member position.

Applies to

Baseline, Marker

Report Expression

Specifies a report expression.

Applies to

As of Time Expression

Report Expression

Specifies the expression to evaluate when determining the value for this variable.

Applies to

Variable

Report Expression

Specifies the numeric position from a report expression.

Applies to

Marker

Required

Specifies whether the prompt is required or optional. If this property is set to required, the prompt must have a value entered before the report can be run. The Usage setting of the associated parameterized filter for this property will override the setting of this object. If you edit this property, but do not get the expected results, verify the Usage setting of the associated filter.

For example, if you created this prompt control with the prompt wizard and set the associated parameterized filter to be optional, changing the value of this property is not sufficient to change this setting. You must also edit the filter's Required property to match the setting for this object's Required property.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

Applies to

Date & Time Prompt, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Right Position (px)

Specifies the pixel position of the right edge of the legend.

Applies to

Legend

Rollup Aggregate Function

Specifies the type of aggregation to apply to summarized values. These values appear at the higher levels of lists and crosstabs.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Rollup Processing

Specifies where to compute aggregates. The Extended setting means that aggregates are computed using an extended aggregate operation. The Database setting means that aggregates are computed by the database software. The Local setting means that aggregates are computed by the data retrieval software in the report server, using a running aggregate.

Applies to

Query

Root Members Only

Specifies whether the set contains the root members or all of the members of the hierarchy.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, Hierarchy Set, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Rotate Values

Controls whether or not the values displayed on the chart are rotated when the Chart Orientation property is set to Horizontal. May help make values easier to read on horizontal charts.

Applies to

Combination Chart, Pareto Chart, Progressive Chart

Row Coordinate

Uniquely identifies the row of a node member or spacer on an edge of the crosstab. You cannot modify this value.

Applies to

Crosstab Intersection

Rows Per Page

Specifies the maximum number of rows to show at one time.

Applies to

Crosstab, List, Repeater, Select & Search Prompt, Tree Prompt, Value Prompt

Scale

Specifies whether the numeric scale is logarithmic or linear.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

Scale Interval

Specifies the interval between ticks on the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

Scope

Specifies the scope of the filter in terms of the number of levels.

Applies to

Summary Filter

Select UI

Specifies which interface the prompt control renders.

Applies to

Date & Time Prompt, Date Prompt, Time Prompt, Value Prompt

Separator

Specifies the separator to use when showing multiple values.

Applies to

Context Item Text, Legend

Series Color

Specifies whether the graphs for a series will have the same color for each combination measure.

Applies to

3-D Combination Chart, Combination Chart

Set Operation

Specifies the set operation to apply to one or more queries, that results in a projection list on which other queries can be based.

Applies to

Query Operation

Set Sorting

Specifies how the set is sorted. By default, the set is not sorted.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, Explicit Member Set, Hierarchy Set, HTML Item, Image, Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Member Children Set, Named Set, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Show Caption

Specifies whether, or where, to show the caption.

Applies to

Field Set

Show Data Range in Legend

Specifies whether to show the full range of data in the legend. If set to No, only the values from the palette will be shown.

Applies to

Point Layer, Region Layer

Show Features with No Data

Indicates whether to show the features of a map that do not have corresponding data.

Applies to

Point Layer, Region Layer

Show Feelers

Specifies whether feeler lines are rendered for each marker.

Applies to

3-D Scatter Chart

Show Legend Values

Specifies whether and how to show legend values.

Applies to

Legend

Size & Overflow

Specifies the height and width of the object, as well as how overflow content should be treated, using scroll bars and clipping.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Text Box Prompt, Text Item, Tree Prompt, Value Prompt

Size Legend Title

Specifies a title within the legend above the palette for the point size. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to

Point Layer

Solve Order

Specifies the solve order in crosstabs and charts. The item with the lowest solve order value is calculated first, followed by the next lowest value, and so on. For identical values, in crosstabs, column items are calculated first, then row items, and then the measure. In charts, x-axis items are calculated first and then legend items.

Applies to

Calculated Measure, Calculated Member, Data Item

Sorting

Specifies the desired sort sequence.

Applies to

Chart Node Member, Crosstab Node Member, Level, Select & Search Prompt, Tree Prompt, Value Prompt

Source Type

Specifies the source type of the text.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Source Type

Specifies the source type of the HTML text.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Source Type

Specifies the source type of the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Source Type

Specifies the source type of the bookmark.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Spacing & Breaking

Specifies text properties such as line height, letter spacing, and word breaking.

Applies to

Block, Class, Conditional Block, Conditional Block, Context Item Text, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Hyperlink, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Break Text, Page Footer, Page Header, Repeater Table Cell, Table Cell, Table of Contents, Table of Contents Entry, Text Item

Spider Effects

Specifies whether the chart is rendered with web-like flat concentric circles. The Radial Axis property must be set to Show for this property to take effect.

Applies to

Polar Chart, Radar Chart

SQL

The text of the typed-in SQL. It is assumed to be appropriate for the type and data source. If it is not the query may fail, or produce unexpected results.

Applies to

SQL

SQL join syntax

Controls the syntax to generate for joins. Click Implicit to generate joins in the WHERE clause. Click Explicit to generate INNER JOIN syntax. If unspecified, the value of the corresponding governor in the model is used.

Applies to

Query

SQL Syntax

Specifies the syntax of the SQL in the query. A value of Cognos indicates that Cognos extended SQL-92 syntax is used. Alternatively, a value of Native indicates that native database SQL is used.

Applies to

SQL

Standard Deviations

Specifies a distance from the mean in standard deviations. This value can be positive or negative. A value of zero indicates the mean value.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker, Marker

Static Choices

Represents a collection of static choices used by the prompt object.

Applies to

Select & Search Prompt, Value Prompt

Style Variable

Specifies a variable based on which the object can be conditionally styled.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Angular Measure, Axis Title, Block, Bubble Chart, Bubble Measure, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Text Item, Chart Title, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab

Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date & Time Prompt, Date Prompt, Default Measure, Display Layer, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Hyperlink Button, Image, Interval Prompt, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Break Text, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Point Layer, Point Measure, Point Size Measure, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Radial Measure, Region Layer, Region Measure, Repeater Table, Repeater Table Cell, Scatter Chart, Select & Search Prompt, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Target Measure, Text Box Prompt, Text Item, Time Prompt, Tolerance Measure, Tree Prompt, Value Prompt, X Axis, X Axis, X Axis Measure, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Y Axis Measure, Z Axis, Z Axis, Z Axis Measure

Subtitle

Specifies whether a chart subtitle is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Suppress

For SAP BW and DB2 OLAP data sources, specifies the type of suppression to apply to the query results. This property overrides the corresponding governor in the model. If unspecified, the value of the governor in the model is used.

Applies to

Query

Table of Contents

Specifies the name of the table of contents to which the entry belongs.

Applies to

Table of Contents Entry

Table of Contents Name

Specifies the name that uniquely identifies the table of contents.

Applies to

Table of Contents

Table Properties

Specifies the properties for the table object.

Applies to

Class, Crosstab, List, Repeater Table, Table

Target Color

Specifies a color for the vertical lines that mark the target ranges for target measure values in a metrics chart.

Applies to

Metrics Range Chart

Target Marker

Specifies whether the status indicators will appear in the legend.

Applies to

Metrics Range Chart

Target Marker Border Color

Specifies a color for the borders around target value markers in a metrics chart.

Applies to

Metrics Range Chart

Target Marker Position

Specifies whether the status indicators will be rendered over the first bar in the cluster or the middle of the cluster. Does not apply to stacked charts.

Applies to

Metrics Range Chart

Target Range (%)

Specifies target ranges centered around target measure values.

Applies to

Metrics Range Chart

Text

Specifies the static text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Text Flow & Justification

Specifies text flow properties, such as direction, writing mode, and justification.

Applies to

Block, Class, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Hyperlink, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page, Page Body, Page Break Text, Page Footer, Page Header, Repeater Table, Repeater Table Cell, Table, Table Cell, Table of Contents, Table of Contents Entry, Text Item

Text Source Variable

Specifies a variable based on which the text source can be chosen.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Title

Specifies whether a chart title is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Tolerance Color

Specifies a color for the vertical lines that mark the tolerance ranges for target measure values in a metrics chart.

Applies to

Metrics Range Chart

Tolerance Label

Specifies whether the label for the Target Tolerance will appear in the legend.

Applies to

Metrics Range Chart

Tooltips

Specifies whether tooltips are rendered in the chart. Tooltips are not supported in PDF documents.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Top Position (px)

Specifies the pixel position of the top edge of the legend.

Applies to

Legend

Total Column

Specifies whether a total column is rendered.

Applies to

Progressive Chart

Total Column Color

Represents the color, gradient, or pattern of the total column on the progressive chart.

Applies to

Total Column

Truncation

Specifies whether labels can be truncated.

Applies to

Gauge Labels, Ordinal Axis, Pie Labels, X Axis, Y Axis

Truncation Text

Specifies the text to append when a label is truncated.

Applies to

Gauge Labels, Legend, Ordinal Axis, Pie Labels, X Axis, Y Axis

Type

Specifies the type of variable.

Applies to

Variable

Type

Specifies the behavior of the prompt button.

Applies to

Prompt Button

Type

Specifies the type of object.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Calculated Measure, Calculated Member, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Explicit Member Set, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level Set, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Measure, Member, Member Children Set, Member Property, Named Set, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Set Expression, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Upper Range Skew (%)

Specifies a percentage that affects the positioning of tolerance bar and range with respect to the target value.

Applies to

Metrics Range Chart

URL

Specifies the URL, using static text.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

URL Source Variable

Specifies a variable based on which the URL source can be chosen.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Usage

Specifies whether the usage of this object is Optional, Required, or Disabled. When Optional, this condition applies if all parameters referenced by the expression are provided with values. If the expression does not refer to any parameters, then this condition is always applied. When Disabled, this condition is never applied, which is useful for authoring and testing purposes.

Applies to

Detail Filter, Summary Filter

Use 1.x Behavior

Specifies that Cognos ReportNet query semantics are used if they differ from Cognos 8 query rules.

Applies to

Query

Use Detail Value on Page

Specifies whether to render a detail value or an aggregate value for a text item that uses a data item as its source. Use this property only when you want to render the value that appears in the first or last detail row of a list, repeater or repeater table on the same page as the text item.

Applies to

Text Item

Use for Parameter Info

Specifies whether the query should be given priority when determining parameter information. Queries with this property set to Yes are checked for parameter information first, followed by queries with this property set to Default. Queries with this property set to No will not be checked for parameter information unless it is referenced in a query that will be checked. Setting this property to Yes on parameterized queries can improve performance in displaying prompt pages.

Applies to

Query

Use Local Cache

Specifies whether a query is a candidate for query reuse. If set to Yes, the query engine can reuse an existing SQL result. If set to No, the query is executed rather than using cached results.

Applies to

Query

Use Same Range For All Instances

Specifies that all instances of the chart use the same maximum value. When set to No, the axis maximum value is recalculated for each chart instance. It is only relevant if the chart is involved in a master detail relationship.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Z Axis, Z Axis

Use SQL With Clause

Specifies whether to send a request to the database using an SQL WITH clause. When set to Yes, and if the database supports WITH clauses, a WITH clause request is generated. When set to No, or if the database does not support WITH clauses, a request using derived tables is generated.

Applies to

Query

Use Thousands Separator

Specifies whether to delimit digit groups with the thousands separator.

Applies to

Text Box Prompt

Use Value

Specifies the values used by the prompt object.

Applies to

Tree Prompt

Use Value

Specifies the values used by the prompt object. These values can be different than the ones that are rendered to the user.

Applies to

Select & Search Prompt, Value Prompt

Value Location

Specifies where values and labels are to be rendered in the chart.

Applies to

Area, Bar, Bubble Chart, Combination Chart, Cumulation Line, Line, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Value Representation

Specifies whether values are rendered as percentages.

Applies to

Pie Chart

Values

Specifies what values to show in the chart and whether to show the corresponding measure, series, or category label.

Applies to

3-D Scatter Chart, Bubble Chart, Polar Chart, Scatter Chart

Values

Specifies whether values are rendered in the chart.

Applies to

3-D Area, 3-D Bar, 3-D Line, Area, Bar, Cumulation Line, Line, Pareto Chart, Point Layer, Progressive Chart, Radar Chart, Region Layer

Values

Specifies whether values are rendered.

Applies to

Pie Chart

Value Type

Specifies whether absolute values are rendered rather than cumulative values.

Applies to

Area, Bar, Line

Vertical Alignment

Specifies how objects contained in this object are vertically aligned.

Applies to

Class, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Footer, Page Header, Repeater Table Cell, Table Cell, Table Row

Visible

Specifies whether to display the object. When set to No, the object is hidden but a fixed space is reserved in the report.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Cumulation Line Axis, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Image, Interval Prompt, Legend, Legend Title, List, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page Break Text, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Radar Chart, Radial Axis, Repeater Table, Scatter Chart, Select & Search Prompt, Table, Table of Contents, Table of Contents Entry, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

Visual Angle

Specifies the angle, in degrees, in which the chart objects will be displayed when the chart has 3-D effects.

Applies to

Combination Chart, Metrics Range Chart, Pareto Chart, Progressive Chart

White Space

Specifies how white space inside the object is handled.

Applies to

Block, Class, Conditional Block, Conditional Block, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Footer, Page Header, Repeater Table Cell, Table Cell, Table of Contents, Table of Contents Entry

Width (px)

Specifies the width of the note, in pixels.

Applies to

Note

X Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

Y1 Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart, Metrics Range Chart

Y2 Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart

Y2 Axis Position

Specifies how the second Y axis is rendered.

Applies to

Combination Chart

Y Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

Data Formatting Properties

The following is a list of properties available in the data formatting dialog.

"Not Applicable" Characters

Specifies the characters to be displayed when the value to be formatted was not applicable. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

Any Error Characters

Specifies the characters to be displayed when the value to be formatted was not available because of an error. This property is overridden by the more specific formatting error conditions, such as Security Error Characters. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

Calendar Type

Specifies the type of calendar to be displayed. The date values will be mapped to the selected calendar before being formatted. The default value is inherited from the user's content language. Note that the Japanese Imperial setting is only applicable for Japanese languages.

Clock

Specifies whether to display the time in 12-hour or 24-hour format. The default value is inherited from the user's content language.

Currency

Specifies the currency to be used. The default currency symbol will be displayed unless the values of the Currency Display and Currency Symbol properties are changed. The default value is inherited from the model.

Currency Display

Specifies whether to display the international or local currency symbol. By default, the local currency symbol is displayed.

Currency Symbol

Specifies a character or characters to use as the symbol to identify the local currency. This symbol will precede the number and any sign, even if it is a leading sign. A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

Currency Symbol Position

Specifies where the currency symbol will appear. If End is selected, any spaces that follow the character or characters in the Currency Symbol or International Currency Symbol properties will be rendered between the number and the symbol. The default value is inherited from the user's content language.

Date Ordering

Specifies the order in which to display the day, month, and year. The default value is inherited from the user's content language.

Date Separator

Specifies the character to be displayed between the year, month, and day. The default value is inherited from the user's content language.

Date Style

Specifies the date style. The results rendered are determined by the language. Generally, Short uses only numbers, Medium uses some abbreviated words, Long uses complete words, and Full includes all available details.

Decimal Separator

Specifies the character that will separate non-decimal numbers from decimals. This property is ignored if no decimals are displayed. The default value is inherited from the user's content language.

Display AM / PM Symbols

Specifies whether to display the AM or PM symbols. The default value is inherited from the user's content language.

Display As Exponent

Specifies whether to render values in scientific notation, using exponents. If this property is set to No, scientific notation will not be used. If this property is not specified, scientific notation will be used only when values exceed the maximum number of digits. The default value is inherited from the user's content language.

Display Days

Specifies whether to display the day. The format of the day can be controlled by selecting one of the specific formats. Selecting Julian means that the 3-digit day of the year will be displayed. The default value is inherited from the user's content language.

Display Eras

Specifies whether to display the era. The default value is inherited from the user's content language.

Display Hours

Specifies whether to display the hours. The default value is inherited from the user's content language.

Display Milliseconds

Specifies whether to display the milliseconds. The format of the milliseconds can be controlled by selecting one of the specific formats. This property is ignored if seconds are not displayed. The default value is inherited from the user's content language.

Display Minutes

Specifies whether to display the minutes. The format of the minutes can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

Display Months

Specifies whether to display the month. The format of the month can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

Display Months

Specifies whether to display the month.

Display Seconds

Specifies whether to display the seconds. The format of the seconds can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

Display Time Zone

Specifies whether to display the time zone. The default value is inherited from the user's content language.

Display Weekdays

Specifies whether to display the weekday. The format of the weekday can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

Display Years

Specifies whether to display the year. The first two digits of the year, which indicate the century, can be controlled by selecting one of the associated property values. The default value is inherited from the user's content language.

Display Years

Specifies whether to display the year.

Divide By Zero Characters

Specifies the characters to be displayed when a numeric value is the result of a division by zero. The default value is /0. Note that the format will be applied only if the data source supports this error condition.

Exponent Symbol

Specifies the character to be displayed to identify exponents if the scientific notation is used. The symbol will be rendered after the number, separated by a space. The default value is inherited from the user's content language.

Group Size (digits)

Specifies the primary grouping size. If a value is specified it represents the number of digits to the left of the decimal point to be grouped together and separated by the thousands separator. The default value is inherited from the user's content language.

International Currency Symbol

Specifies a character or characters to use as a symbol to identify the international currency. This symbol will replace the currency symbol. A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

Mantissa (digits)

Specifies the number of digits to be displayed following the exponent symbol if the scientific notation is used.

Maximum No. of Digits

Specifies the maximum number of digits that can be displayed. If the maximum number of digits is not sufficient to display the value, a scientific notation will be used. The default value is inherited from the user's content language.

Minimum No. of Digits

Specifies the minimum number of digits that can be displayed. If the minimum number of digits is too high to display a value, the padding character will be used. The default value is inherited from the user's content language.

Missing Value Characters

Specifies the character or characters to be displayed when the value is missing. If no value is entered for this property, an empty string will be displayed.

Negative Pattern

Specifies a presentation format, based on patterns, for negative numbers. Some restrictions exist. The numerical part of the negative pattern is ignored. Only the suffix and the prefix are used. For example, in the pattern ABC#,##0.#EFG, ABC is the prefix, EFG is the suffix and #,##0.# is the numerical part of the pattern.

Negative Sign Position

Specifies where the negative sign will appear. The default value is inherited from the user's content language.

Negative Sign Symbol

Specifies how to display negative numbers. The default value is inherited from the user's content language.

No. of Decimal Places

Specifies the number of digits to be displayed to the right of the decimal point. If this property is not set, the number of decimal places will vary depending on the number rendered.

Numeric Overflow Characters

Specifies the characters to be displayed when a numeric value is the result of a numeric overflow. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

Padding Character

Specifies the character that will be used to pad values that have fewer digits than the minimum number of digits. The default value is inherited from the user's content language.

Pattern

Specifies a presentation format that is based on patterns.

Percentage Symbol

Specifies whether to display the values per hundred (percent) or per thousand. The symbol will be appended to the number and any trailing sign. A space between the numeric value and the symbol can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

Percent Scale (integer)

Scale to be applied to value after formatting. If omitted, no percent scale will be applied and the value will formatted according the normal decimal positioning associated with the percent (or per mille) symbol.

Scale

Specifies how many digits to move the decimal delimiter for formatting purposes. For example, move the decimal three spaces to present values in thousands. The default value is inherited from the database field.

Secondary Group Size (digits)

Specifies the secondary grouping size. If a value is specified it represents the number of digits to the left of the primary group that will be grouped together and separated by the thousands separator. If this property is left blank, the secondary grouping of digits is the same number as the primary group size, as specified by the Group Size (digits) property. The default value is inherited from the user's content language.

Security Error Characters

Specifies the characters to be displayed when the value to be formatted was not available for security reasons. The default value is #!Security. Note that the format will be applied only if the data source supports this error condition.

Thousands Separator

Specifies how to delimit digit groups, such as thousands. This property is only used if the Use Thousands Separator property is set to Yes. The default value is inherited from the user's content language.

Time Separator

Specifies the character to be displayed between the hour, minute, and second. The default value is inherited from the user's content language.

Time Style

Specifies the time style to be displayed. The exact results that will be rendered are determined by the language. Generally, Short means that the minimum details will be displayed, Long adds seconds, and Full means that all details are displayed, including the time zone. The default value is inherited from the user's content language.

Time Unit

Specifies the unit of measure of the value. This property will be ignored if any day or time components are shown. The default value is inherited from the user's content language.

Use Thousands Separator

Specifies whether the grouping delimiter will be applied as defined by the Group Size property. The default value is inherited from the user's content language.

Zero Value Characters

Specifies the character or characters to be displayed when the value is zero (0). If no value is entered for this property, the Maximum No. of Digits property determines how many zero digits are displayed.

Appendix H: Using Patterns to Format Data

You can format data so that it matches any pattern of text and numbers when default formats are not appropriate. For example, you can format dates to use full text including the era, or you can format them to only use numbers and show the last two digits of years to save space.

Using symbols and patterns can provide similar results as basic data formatting tasks. For example, you can set how many digits appear after the decimal point. You can achieve these types of results with a pattern, or you can set the No. of Decimal Places property. Patterns allow flexibility for more complex requirements.

Each supported content language code requires a specific set of symbols to be used in patterns. For each language code, there are two tables you will need; one for date and time symbols, and one for decimal symbols. The decimal symbols are the same for all locales, however, date and time symbols are grouped into six locale groups. Check the Date and Time Symbol section to see which locale group is used for your locale.

To define patterns, open the **Data Format** dialog box, and edit the **Pattern** property for each format type. Use the symbols that are defined in the language code tables, and follow these guidelines.

Pattern Guidelines

When you define a pattern, the number of symbols you use affects how the data will be shown. There are different rules for text, numbers, and values that can take the form of text or numbers.

Text

You can specify whether text is produced in full or abbreviated form.

Number of symbols	Meaning	Example
4 or more	Full text form	EEEE produces Monday
Less than 4	Abbreviated form	EEE produces Mon

Numbers

The number of symbols you use in a pattern sets the minimum number of digits that are produced in a report. Numbers that have fewer digits than specified are zero-padded. For example, if you specify mm for minutes, and the database value is 6, the report will show 06.

Note: The year value is handled differently. If you specify two symbols for year, the last two digits of the year value is produced. For example, yyyy produces 1997, and yy produces 97.

Text and Numbers

For values that can produce text or numbers, such as months, you can specify whether text or numbers are produced, and whether words are abbreviated.

Number of symbols	Meaning	Example
3 or more	Text	MMMM produces January MMM produces Jan
Less than 3	Numbers	MM produces 01 M produces 1

Date and Time Symbols

Locale ID	Locale group
af-za, en, en-au, en-be, en-bw, en-ca, en-gb, en-hk, en-ie, en-in, en-mt, en-nz, en-ph, en-sg, en-us, en-vi, en-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, ms-my, nb-no, nl, nl-be, nl-nl, no, no-no, om-et, om-so, pl, pl-pl, pt, pt-br, pt-pt, so-dj, so-et, so-ke, so-so, sv, sv-fi, sv-se, sw-ke, sw-tz	Locale Group A

Locale ID	Locale group
be-by, bg-bg, el, el-gr, fi, fi-fi, hr, hr-hr, hu, hu-hu, ja, ja-jp, ko, ko-kr, ro, ro-ro, ru, ru-ua, ru-ru, sh-yu, sk, sk-sk, sl-si, sq-al, sr-sp, th, tr, tr-tr, uk-ua, zh, zh-cn, zh-hk, zh-mo, zh-sg, zh-tw	Locale Group B
ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk	Locale Group C
de, de-at, de-be, de-ch, de-de, de-lu	Locale Group D
fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu	Locale Group E
ga-ie	Locale Group F

Locale Group A

Locales: af-za, en, en-au, en-be, en-bw, en-ca, en-gb, en-hk, en-ie, en-in, en-mt, en-nz, en-ph, en-sg, en-us, en-vi, en-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, ms-my, nb-no, nl, nl-be, nl-nl, no, no-no, om-et, om-so, pl, pl-pl, pt, pt-br, pt-pt, so-dj, so-et, so-ke, so-so, sv, sv-fi, sv-se, sw-ke, sw-tz

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	y	Number	1996
Year (of 'Week of Year')	Y	Number	1996

Meaning	Symbol	Presentation	Example
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	d	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	k	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	h	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group B

Locales: be-by, bg-bg, el, el-gr, fi, fi-fi, hr, hr-hr, hu, hu-hu, ja, ja-jp, ko, ko-kr, ro, ro-ro, ru, ru-ua, ru-ru, sh-yu, sk, sk-sk, sl-si, sq-al, sr-sp, th, tr, tr-tr, uk-ua, zh, zh-cn, zh-hk, zh-mo, zh-sg, zh-tw

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	a	Number	1996
Year (of 'Week of Year')	A	Number	1996
Month in year	n	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	j	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	x	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55

Meaning	Symbol	Presentation	Example
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group C

Locales: ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	u	Number	1996
Year (of 'Week of Year')	U	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	t	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	h	Number	24

Meaning	Symbol	Presentation	Example
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group D

Locales: de, de-at, de-be, de-ch, de-de, de-lu

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	j	Number	1996
Year (of 'Week of Year')	J	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	t	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)

Meaning	Symbol	Presentation	Example
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group E

Locales: fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	a	Number	1996
Year (of 'Week of Year')	A	Number	1996
Month in year	M	Text and number	July and 07

Meaning	Symbol	Presentation	Example
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	j	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	x	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group F

Locales: ga-ie

Meaning	Symbol	Presentation	Example
Era	R	Text	AD
Year	b	Number	1996
Year (of 'Week of Year')	B	Number	1996
Month in year	M	Text and number	July and 07
Week in year	t	Number	27
Week in month	T	Number	2
Day in year	l	Number	189
Day in month	L	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	u	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	U	Number	0
Minute in hour	n	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	c	Text	Pacific Standard Time

Meaning	Symbol	Presentation	Example
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Decimal Format Symbols

All locales

Symbol	Meaning
0	A digit that is shown even if the value is zero.
#	A digit that is suppressed if the value is zero.
.	A placeholder for decimal separator.
,	A placeholder for thousands grouping separator.
E	Separates mantissa and exponent for exponential formats.
;	Separates formats for positive numbers and formats for negative numbers.
-	The default negative prefix.
%	Multiplied by 100, as percentage.
‰	Multiplied by 1000, as per mille.
¤	The currency symbol. If this symbol is present in a pattern, the monetary decimal separator is used instead of the decimal separator.
¤¤	The international currency sign. It will be replaced by an international currency symbol. If it is present in a pattern, the monetary decimal separator is used instead of the decimal separator.
X	Other characters that can be used in the prefix or suffix.
'	Used to quote special characters in a prefix or suffix.
/u221E	Infinity symbol.
/uFFFD	Not a Number symbol.

Appendix I: Drilling Through from Cognos Series 7 to Cognos 8

You can set up drill-through access from Cognos Series 7 to Cognos 8. Specifically, you can drill through to a Cognos 8 report from a PowerPlay Web report or Cognos Visualizer report. Drill through is supported for both PowerCubes and third-party cubes.

You must complete steps in both Cognos Series 7 and Cognos 8 to enable drill through. Refer to the Cognos Series 7 documentation for the specific steps you must complete in the Cognos Series 7 components. Cross references to the appropriate Cognos Series 7 documents are included below.

Setting Up Drill-through Access from PowerPlay Web

Setting up drill-through access from PowerPlay Web to Cognos 8 involves

- for PowerCubes, specifying drill-through targets for Cognos 8 reports in the Transformer model

  For more information, see the Transformer documentation.

- for third-party cubes, specifying drill-through targets for Cognos 8 reports in PowerPlay Connect.

  For more information, see the PowerPlay OLAP Server Connection Guide.

- configuring drill-through access in PowerPlay Server Administration

  In addition to enabling drill-through access to Cognos 8, you must specify the location of the Cognos 8 server and the Cognos 8 folder that contains the target reports. For more information, see the PowerPlay *Enterprise Server Guide*.

- selecting the filters to add to the target report

  In PowerPlay Enterprise Server Administration, enable and use Cognos 8 Assistance to identify the filter expressions required in the target report. For more information, see the PowerPlay *Enterprise Server Guide*.

- creating and testing the target report (p. 654)

Setting Up Drill-through Access from Cognos Visualizer

Setting up drill-through access from Cognos Visualizer to Cognos 8 involves

- specifying the Cognos 8 target and selecting the filters to add to the target report

  You must configure drill through to Cognos 8 for individual Cognos Visualizer reports. For more information, see the Cognos Visualizer *User Guide*.

- creating and testing the target report (p. 654)

Create and Test the Target Report

The target report must be based on a published package that contains the metadata items that you want to filter on, or contains items that are mapped to those metadata items.

When you create the target report, ensure that the names of the parameters you add are identical to the parameter names listed in the **Drill Through Assistant** page in Cognos Series 7. However, the metadata item that you use in the target report for that parameter name does not have to be the identical label. The data values between the target parameter and the source value shown in the drill assistant must match. You may also need to change the type of operator in the target parameter from what is recommended in the **Drill Through Assistant**. For example, if the assistant recommends an = operator but you want to pass a date range, you should change the parameter operator in the target to `in_range`.

Steps

1. Start Report Studio and create a new report.
2. Add the data items and other objects you want.
3. From the **Data** menu, click **Filters** .
4. In the **Detail Filters** tab, click the add button.
5. In the **Expression Definition** box, create the parameterized filter you want by typing the filter expression.
6. Click **OK**.
7. In the **Usage** box, click **Optional**.

   If you do not make the filter optional, a prompt page appears when you drill through to the report.
8. Repeat steps 4 to 7 for other parameterized filters you want to add.
9. Save the report.

   The report name must match what you specified as a target in the PowerCube, third-party cube, or Cognos Visualizer report.
10. Test the drill through in the PowerPlay report or Cognos Visualizer report.

Glossary

burst

To create many report results by running a single report once. For example, you can create a report that shows sales for each employee, and run it once, sending different results to regional managers by bursting on region. You set up bursting in Report Studio and enable it in the portal.

calculated member

A member of a dimension whose measure values are not stored but are calculated at run time using an expression.

cardinality

For OLAP data sources, cardinality is the number of members in a hierarchy. The cardinality property for a hierarchy is used to assign solve orders to expressions. For relational data sources, cardinality indicates the nature of the relationship between two query subjects, query items, or other model objects.

cascading prompt

A prompt that uses values from a previous prompt to filter the values in the current prompt or picklist.

condition

An expression that yields a boolean value. Conditions are used in query expressions, query filters, and boolean report variables that can be used for conditional formatting, styles, data sources, layouts and blocks.

Content Manager

The Cognos 8 service that manages the storage of customer applications, including application-specific security, configuration data, models, metrics, reports, and report output. Content Manager is needed to publish models, retrieve or store report specifications, manage scheduling information, and manage the Cognos namespace.

cube

A physical data source containing a multidimensional representation of data. A cube contains information organized into dimensions and optimized to provide faster retrieval and navigation in reports. In Cognos Planning, a cube (see also D-Cube) corresponds to a tab on Contributor client user interface.

data source

A relational database, dimensional cube, file, or other physical data store that can be accessed though Cognos 8.

data tree

Within a studio, contains objects such as query subjects, query items, dimensions, levels, and members. A data tree is used as a palette of the available data that can be inserted into calculations, filters, display areas, and other authoring gestures.

dimension

In Cognos Planning, the rows, columns, and pages of a cube are created from dimensions. Dimensions are lists of related items such as Profit and Loss items, months, products, customers, and cost centers. Dimensions also contain all the calculations. One dimension can be used by many cubes.

In Cognos 8 BI a dimension is a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets. Each dimension includes different levels of members in one or more hierarchies and an optional set of calculated members.

drill down

In Cognos Planning, drill down is a technique used to analyze D-Cube data that was imported by a D-Link. You can drill down on any single cell in a D-Cube. If the cell contains data transferred by a D-Link, drill down opens a view of the source data. If the data was imported from another D-Cube, drill down opens the appropriate selection from the source D-Cube. If the data was imported from an external source (a mapped ASCII file or an ODBC database), drill down extracts the relevant data from the source file and displays it in a special drill-down results dialog box.

In Cognos 8 BI, drill down refers to the act of navigating from one level of data to a more detailed level. The levels are set by the structure of the data. See also drill up.

drill up

The act of navigating from one level of data to a less detailed level. The levels are set by the structure of the data.

See also drill down.

group

In security, a list of users or other groups that can be used to assign access permissions and capabilities.

Groups can be referenced from third-party authentication sources or can be local to Cognos 8. Local groups are managed from the administration portal. The list of groups that an authentication user is a member of is part of the user's passport for a Cognos 8 session.

In reporting, grouping is the action of organizing common values of query item together and only displaying the value once. Headers and footers often appear after each instance of a common value in a grouped column

hierarchy

A hierarchy represents a collection of dimensional members organized into a tree structure, with each member having one or more parent members and an arbitrary number of child members.

The root of a hierarchy has no parent, and leaf members of a hierarchy have no children.

layout

In reporting, layout defines the appearance of the report, including formatting, style, and design.

In report specifications, layout is the portion of the specification that defines how the data returned by queries is presented.

level

A level is a set of members that have common attributes. For example, a geographical dimension might contain levels for country, region, and city. Levels are used to define the structure of hierarchies in a dimension.

locale

A code that is used to set the language or dialect used for browsers, report text, and so on; and the regional preferences, such as formats for time, date, money, and money expressions.

For Cognos products, you can specify a locale for the product interface (product locale) and for the data in the report (content locale).

MDX

An abbreviation for multidimensional expression language, MDX is the multidimensional equivalent of SQL. Cognos 8 allows report authors to type in their queries using MDX expressions in Report Studio for dimensional data sources.

measure

A performance indicator that is quantifiable and used to determine how well a business is operating. For most business purposes, the aggregate values of a measure are more useful than individual values. For example, measures can be Revenue, Revenue/Employee, and Profit Margin %. In relational modeling, this is also called a "fact."

member

A member is a unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products hierarchy.

See also member unique name.

Metric Studio application

A set of metric types, metrics, and scorecards that monitor the performance of an organization. Each organization may have several Metric Studio applications monitoring distinct units, such as operating companies or divisions that have different user communities, metric types or reporting calendars.

model

A physical or business representation of the structure of the data from one or more data sources. A model describes data objects, structure, and grouping, as well as relationships and security.

In Cognos 8 BI, a design model is created and maintained in Framework Manager. The design model or a subset of the design model must be published to the Cognos 8 server as a package for users to create and run reports.

In Cognos Planning, a model is a group of D-Cubes, D-Lists, D-Links, and other objects stored in a library. A model may reside in one or more libraries, with a maximum of two for Contributor.

package

A subset of a model, which can be the whole model, to be made available to the Cognos 8 server.

For Metric Studio users, see metric package.

page set

A set of pages associated with a specified group or level that indicates the page set is to be repeated for that group or level. For example, render a specified set of pages for each customer in the query.

prompt

A report element that asks for parameter values before the report is run.

properties pane

Within a studio, the properties pane provides an overview of the properties for selected data. You can also use the properties pane to make several changes and apply them at the same time, instead of repeating several different commands.

query

A specification for a set of data to retrieve from a data source. A report specification can contain one or more queries.

The type of object created and edited by Query Studio. A query is a subtype of report.

query item

A representation of a column of data in a data source. It contains a reference to a database column, a reference to another query item, or a calculation. Query items may appear in a model or in a report.

query subject

One of the types of objects inside a model. A query subject can be defined as a collection of references to items of other query subjects, or expressed as an SQL expression that represents selected query items. These query items will be retrieved from objects such as tables, synonyms, views, and so on. Query subjects contain query items.

repeater

In Report Studio, a cell container that repeats values within itself with no predefined internal structure.

repeater table

In Report Studio, a table-like container that repeats cells across and down the page for row in the associated query.

report

A set of data deliberately laid out to communicate business information. Depending on the context, "report" may refer to report specification or report output.

Report (more precisely, report specification) is the generic term for the objects created or edited by Query Studio, Report Studio, and Analysis Studio.

report output

Report output combines data at the point in time when the report was run with a report specification. It is a document that can be displayed, printed, or emailed. Cognos 8 can produce report outputs in HTML, PDF, Excel, or CSV formats.

report specification

The definition of queries, prompts, layouts, and styles that make up a report. A report specification is combined with data by a run operation to create report outputs. You create report specifications by using Report Studio, Query Studio, Analysis Studio, or through the Software Development Kit.

style sheet

A file that defines the formatting and appearance of data or a document. In XML, style sheets may be extensible style sheet language (XSL) files or extensible style sheet language transformation (XSLT) files.

In HTML, style sheets are cascading style sheet (CSS) files.

XSL and CSS style sheets can be embedded inside any XML document or referenced as an external, separate file.

summary

In reporting and analysis, summaries are aggregate values that are calculated for all the values of a particular level or dimension. Examples of summaries include total, minimum, maximum, average, and count.

template

A reusable report layout or style that can be used to set the presentation of a query or report.

work area

The area within a studio that contains the report, analysis, query or agent currently being used.

Index

Symbols

"Not Applicable" Characters, 633
100% stacked charts, 348
3-D Area, 523
3-D Bar, 523
3-D charts, 349
3-D Combination Chart, 524
3-D Line, 524
3-D Scatter Chart, 524
3-D Viewing Angle, 564

A

absolute charts, *See* standard charts
Absolute Position, 564
Across, 564
adding
    bookmarks, 141
    borders, 137
    calculations, 189
    colors, 138
    data, 64
    footers, 135
    headers, 135
    languages, 265
    layouts, 264
    multimedia files, 143
    pages, 151
    prompt buttons, 211, 212
    prompts, 199
    query macros, 231
    summaries, 182
    text, 137
    variables, 259
aggregate button, 184
Aggregate Function, 565
aggregate functions
    using in data item expressions, 185
aggregating
    member sets, 188
aggregating values
    charts, 186
    crosstabs, 186
aggregation
    samples and time dimensions, 334
aggregation modes, 57
aggregation properties
    model, 183
aliases
    in maps, 122
aligning
    objects, 147
Allow 45° Rotation, 565
Allow 90° Rotation, 565
Allow Skip, 565
Allow Stagger, 565
Analysis Studio analyses
    opening, 236
Angular Axis, 524, 566
Angular Measure, 525
Any Error Characters, 633
Application, 566
application error when upgrading a report, 317
applying
    padding, 148
    table styles, 147
Apply Single Class, 566
Area, 525
area charts, 342
As of Time Expression, 525
authoring modes, 61
Auto Group & Summarize, 183, 566
automating
    Cognos 8 using the SDK, 78
Auto-Sort, 567
Auto-Submit, 567
Auto Truncation, 567
Avoid Division by Zero, 567
Avoid Label Collision, 567
axes
    bipolar, 293

charts, 337
customizing titles, 115
dual, 293
title in maps, 126
Axis Assignment, 567, 568
Axis Labels, 568
Axis Line, 568
Axis Title, 525, 568

B

background color
adding, 138
does not appear in template, 317
Background Color, 568
Background Image, 569
background images, 139
backgrounds in charts, 318
Bar, 525
Bar, Line, or Area Index, 569
bar charts, 94, 341
Baseline, 526
baselines
chart, 102
Baselines, 569
bipolar axis, 293
Block, 527
blocks
inserting, 143
Block Variable, 569
Bookmark, 527
bookmarks
adding, 141
creating for drill-through access, 269
Border, 570
Border Color, 570
borders
adding, 137
formatting, 137
Borders, 570
Bottom Position (px), 570, 571
Box Type, 571
breaking
creating page breaks, 181
browsing data
in expression editor, 359
Bubble Chart, 527
bubble charts, 344
Bubble Measure, 527
Build Prompt Page tool, 200
burst groups
defining, 249
bursting
enabling, 251
burst options
setting, 249
burst recipients
creating calculated fields, 247
defining, 246
burst report
cannot view, 321
burst reports
creating, 246
creating against dimensional data sources, 251
bursts
definition, 655
burst tables
creating, 247
business keys
using for drill-through, 271

C

calculated fields
creating for bursting, 247
Calculated Measure, 527
calculated measures, 190
Calculated Member, 528
calculated members, 190
definition, 655
Calculation Intersection, 193, 571
calculations
adding, 189, 190
inserting, 143
position, 45
resolving in crosstabs and charts, 191
resolving in dimensional data sources, 193
calculations, member, 102
Calendar Type, 572, 634
CAMID, *See* Cognos Access Manager ID
Caption, 528, 572
cardinality
definition, 655
Cardinality, 572

Cascade Source, 572
cascading prompts
adding, 212
definition, 655
cascading style sheets
classes, 164
Case Insensitive, 572
Catalog, 573
changed features
data formats for OLAP, 45
data tree for SAP BW, 47
equality filters, 47
for aggregate, 46
slicers, 44
summaries on non-additive measures for SAP BW, 46
changing
chart scale intervals, 105
package connections, 243
profiles, 62
referenced objects, 150
user profiles, 62
chart background
changing, 101
Chart Body, 528
chart configurations
100% stacked, 348
3-D, 349
absolute,, 347
changing, 290
stacked, 348
standard, 347
See standard charts
Chart Footer, 528
Chart Node Member, 529
Chart Orientation, 573
charts, 94, 337
adding data, 289
adding notes, 107
aggregating values, 186
area, 342
axis titles, 115
bar, 341
baselines, 102
bubble, 344
changing colors, 99
changing configuration, 290
changing lines, 105
changing presentation, 293
changing the background, 101
circle,, 339
column, 339
combination, 342
combining sets of data, 291
conditional palettes, 117
configurations, 338
converting, 295
creating, 288
curve,, 341
customizing, 96
customizing values, 289
defining query contexts, 115
dial,, 346
display every second label, 318
drill-down, 112
drill-through, 109
drill-up, 112
formatting axis values, 108
gauge, 346
gradient backgrounds appear gray, 318
grouping data, 290
horizontal,, 341
labels overwrite one another, 318
legends, 113
line, 341
markers, 106
metric range, 346
palettes, 99, 100
Pareto, 340
pie, 339
point, 344
polar, 345
progressive column, 340
properties, 96
properties not supported in Microsoft Excel, 520
quadrant, 345
radar, 343
resizing, 161
resolving multiple calculations, 191
scales, 105
scatter, 343
sector,, 339

showing values, 289
spider,, 343
stacked, 94
star,, 343
types, 337
types not supported in Microsoft Excel, 521
vertical bar,, 339
waterfall,, 340
XY,, 343
See bar charts
See column charts
See gauge charts
See line charts
See pie charts
See progressive column charts
See radar charts
See scatter charts
Chart Subtitle, 529
Chart Text Item, 529
Chart Title, 529
Chart Type, 573
choosing
database connections, 229
templates, 63
circle charts, *See* pie charts
Class, 529
classes
changing the default for a report, 164
Classes, 573
Clear Parameter Values, 66
clipboards
copying reports, 236
opening reports, 236
Clock, 634
Clock Mode, 574
Cognos 8 Go! Office
creating reports for, 76
Cognos 8 Software Development Kit, 78
Cognos Access Manager ID, 247
Cognos SQL, 227
Cognos Visualizer reports
setting up drill through to Cognos 8, 653
color, background
does not appear in template, 317
Color Legend Title, 574
colors
adding, 138
changing in charts, 99
custom, 138
column charts, 94, 339
Column Coordinate, 574
columns, 64
adding multiple items in a single column, 243
calculated, 189
changing order, 155
grouping, 80
renaming, 156
setting the group span, 81
sorting, 179
swapping, 157
Column Titles, 574
column widths
Microsoft Excel limitations, 518
Combination Chart, 530
combination charts, 342
combining reports, *See* master detail relationships
Component Override, 530
Component Reference, 575
concatenating
strings, 361
Conditional Block, 530
conditional block lists
inserting, 143
Conditional Palette, 575
conditional palettes, 117
Conditional Styles, 575
Condition Explorer, 52
conditions
defining, 254
definition, 655
configuring
Web browsers, 61
Connecting Lines, 576
Content Manager
definition, 655
Contents Height, 576
context
filtering, 174
Context Item Text, 531
converting
charts, 295

lists to crosstabs, 91
lists to repeaters, 130
reports to templates, 74
to MDX, 227
to SQL, 227
copied objects
updating, 151
copyright material
printing, 38
Count
changes between Cognos ReportNet and Cognos 8, 314
Count Distinct, 314
creating
accessible reports, 75
burst tables, 247
charts, 288
crosstab nodes, 87
crosstabs, 287
dashboards, 303
drill-through reports, 273
drill-up/drill-down reports, 279
filter expressions for drill through from Cognos Series 7 to Cognos 8, 654
lists, 285
mailing labels, 131
master detail relationships, 282
member sets, 169, 281
multilingual reports, 265
multiple-page reports, 301
prompt pages, 201
prompts, 201
Query Studio template, 75
reports, 62
reports for Cognos 8 Go! Office, 76
report specifications, 62
section headings, 180
single-edge crosstabs, 87
tables of contents, 140
target reports for drill through from Cognos Series 7, 654
templates, 74
creating join relationships, 214
Cross Product Allowed, 576
Crosstab, 531
Crosstab Columns, 531
Crosstab Corner, 532
crosstab fact cells
formatting, 89
Crosstab Fact Cells, 532
Crosstab Intersection, 532
crosstab intersections
formatting, 89
crosstab member fact cells
formatting, 89
Crosstab Member Fact Cells, 533
Crosstab Node Member, 533
crosstab nodes, 86
creating, 87
members, 86
moving, 86
crosstab reports, 85
Crosstab Rows, 533, 534
crosstabs
aggregating values, 186
aggregation modes, 57
applying table styles, 147
creating, 287
creating single-edge, 87
crosstab node members, 86
crosstab nodes, 86
discontinuous, 93
formatting, 89
indenting data, 91
intersections, 89
nesting data, 88
overflow error, 321
resolving multiple calculations, 191
Crosstab Space, 534
CSV format
producing reports, 71
cubes
definition, 655
drill through to relational data gives unexpected results, 324
Cumulation Axis, 576
Cumulation Label, 576
Cumulation Line, 534
Cumulation Line Axis, 534
Cumulation Line Label, 534
Cumulative Line, 577

currencies
formatting, 193
Currency, 634
currency data format
Microsoft Excel limitations, 519
Currency Display, 634
Currency Symbol, 634
Currency Symbol Position, 634
Current Block, 577
curve charts, *See* line charts
customizing
charts, 96
maps, 126
Custom Label, 577
custom labels, 111

D

dashboards
creating, 303
using global filters, 199
data
adding, 64
adding to charts, 289
aggregating in charts, 186
aggregating in crosstabs, 186
browsing in expression editor, 359
combining sets in charts, 291
filtering, 172
filtering date columns, 176
filtering SAP BW data source, 175
formatting, 193
formatting using patterns, 641
grouping in charts, 290
highlighting, 257
showing for specific time periods, 197
sorting, 177
summarizing, 182
database connections
choosing, 229
data containers
specify what appears when there is no data, 199
Data Format, 577
data formats
changed features for OLAP, 45
Microsoft Excel limitations, 519
data handling, 40
Data Item, 535, 577, 578
data item expressions
using aggregate functions, 185
Data Item Label, 578
data items
adding, 64
adding multiple items in a single column, 243
browsing data in expression editor, 359
specifying list for an object, 197
Data Item Value, 579, 580
Data Language, 580
data series
charts, 337
Data Source, 580
data sources
conformed dimensions, 215
definition, 655
dimensionally modeled relational, 167, 231, 360
non-conformed dimensions, 215
OLAP, 167, 360
data stores
conformed dimensions, 271
data tree
changes for SAP BW, 47
data trees
customizing, 66
definition, 656
Date, 535
Date & Time Prompt, 535
date and time prompts, 201
Date Ordering, 634
Date Prompt, 535
date prompts, 201
dates
filtering, 176
formatting, 193
using for drill-through access, 272
Date Separator, 634
Date Style, 635
Decimal Separator, 635
Default Measure, 535, 580
default selections
specifying in prompts, 210
Default Selections, 580
Default Title, 581
Define Contents, 581

Define Member Sets, 581
defining
burst groups, 249
burst recipients, 246
deprecation notices, 48
Depth, 581
derived column, 176
Description, 581
Detail, 581
Detail Filter, 536
Diagram Identifier, 582
dial charts, *See* gauge charts
Dial Outline Color, 582
Dictionary, 582
dictionary property in maps, 122
Dimension, 536, 582
dimensional coercion rules, 361
dimensional data sources
creating burst reports, 251
creating group reports, 80
linking members to create a master detail relationship, 283
resolving calculations, 193
running a report against, 69
dimensionally modeled relational data sources, 167, 231, 360
dimensions, 64, 167
conformed for drill-through access, 271
definition, 656
searching, 171
working with, 167
discontinuous crosstabs
creating, 93
Display After Overall Header, 582
Display AM / PM Symbols, 635
Display As Exponent, 635
Display Days, 635
Display Eras, 635
Display Frequency, 582
Display Hours, 635
Display Layer, 536
Display Milliseconds, 583, 635
Display Minutes, 635
Display Months, 636
Display Seconds, 583, 636
Display Time Zone, 636
Display Value, 583
Display Weekdays, 636
Display Years, 636
Divide By Zero Characters, 636
division by zero operation appears different in lists and crosstabs, 317
DMR, *See* dimensionally modeled relational data sources
double counting, 314
Down, 583
drill down
definition, 656
drilling
drill-through links not active in Safari browser, 323
through gives unexpected results, 324
drillIng
between PowerCubes does not work, 323
drilling down
charts, 112
drilling through
charts, 109
maps, 128
Microsoft Excel limitations, 519
drilling up
charts, 112
drill through
setting up for Cognos Visualizer reports, 653
setting up for PowerPlay reports, 653
drill-through access
between packages, 272
between PowerCubes and packages, 272
business keys, 271
concepts, 267, 268
conformed dimentions, 271
formats, 268
members, 269
model-based access, 268
report-based access, 268
using bookmarks, 269
using dates, 272
using scope in model-based, 272
using URLs, 269
values, 269
drill-through definition, 268
Drill-Through Definitions, 583
drill-through reports
creating, 273

specifying text, 276
drill up
definition, 656
drill-up/drill-down reports
creating, 279
member sets, 281
dual axis, 293
Duplicates, 584

E

editing
MDX, 227
SQL, 227
emailing reports
Microsoft Excel format reports, 520
Embed, 584
enabling
bursting, 251
equality filters
changed features, 47
error messages
error occurred while saving the output for the burst instance, 321
HRESULT, 319
ORA-00907, 321
QE-DEF-0288, 320
RSV-SRV-0025, 319
RSV-SRV-0040, 317
UDA-SQL-0114, 319
UDA-SQL-0206, 319
escape character, 359
examples
adding a note to a chart, 107
adding data to charts, 289
adding interval prompts, 204
changing chart configuration, 290
changing chart presentation, 293
changing palettes, 100
combining sets of data in charts, 291
converting charts, 295
create column charts, 95
create drill-through charts, 109
create map reports, 120
create metrics range charts, 297
creating conditional chart palettes, 117
creating discontinuous crosstabs, 93
creating mailing labels, 131
customizing chart palettes, 100
customizing values in chart axis titles, 115
customizing values in charts, 289
defining data values for region layers, 121
formatting axis values, 108
grouping data in charts, 290
showing values in chart legends, 113
showing values in charts, 289
Excel 2000 format
deprecation notices, 48
Execution Method, 584
Execution Optimization, 584
Expand Features, 584
Explicit Member Set, 536
Explorer Bar, 52
Exponent Symbol, 636
Express authoring mode, 61
Expression, 585
expression editor
Block Functions, 396
browsing data, 359
Business Date/Time Functions, 391
Common Functions, 404
Constants, 388
Constructs, 390
creating calculations, 189
creating filters, 172
Data type casting functions, 509
DB2, 423
DB2 Cast, 435
DB2 Math, 436
DB2 Trigonometry, 437
dimensional coercion rules, 361
Informix, 439
Informix Math, 443
Informix Trigonometry, 444
Macro Functions, 397
Member Summaries, 385
MS Access, 445
MS Access Cast, 451
MS Access Math, 452
MS Access Trigonometry, 452
Operators, 365
Oracle, 453
Oracle Math, 460

Oracle Trigonometry, 460
prompts, 205
quotation marks in literal strings, 359
Red Brick, 462
Report Functions, 493
SAP BW, 483
SAP BW Math, 484
SAP BW Trigonometry, 483
searching for values, 359
SQL Server, 468
SQL Server Cast, 473
SQL Server Math, 474
SQL Server Trigonometry, 475
Summaries, 373
Sybase, 484
Sybase Math, 491
Sybase Trigonometry, 491
Teradata, 476
Teradata Trigonometry, 480
expressions
calculations, 190
functions not available, 360
quotation marks in literal strings, 359
extended data items, 58, 164

F

Face Color, 585
Fact, 536
Fact Cells Precedence, 586
facts, 64
Field Set, 537
field sets
inserting, 143
Fill Effects, 586
filter expressions
creating for drill through from Cognos Series 7 to Cognos 8, 654
filtering
data, 172
data from SAP BW data sources, 175
date columns, 176
filtering data, 174
filters
make_timestamp column, 319
slicers, 172
using with multiple-fact queries, 215
finding information, 38
finding objects
in reports, 57
First Column Color, 586
First Date, 586
First Label Index, 586
first-rows optimization, 229
Floating, 162, 587
Font, 587
Font Auto-Sizing, 587
fonts
in maps, 126
specifying, 138
Footer, 587
footers
adding, 135
formatting, 135
in maps, 125, 126
Foreground Color, 588
foreground colors
adding, 138
formats
drill-through access, 268
not supported for Microsoft Excel, 520
reports, 66
formatting
borders, 137
chart axis values, 108
colors, 138
crosstabs, 89
currencies, 193
data, 193
data using patterns, 641
dates, 193
footers, 135
headers, 135
limitations for Microsoft Excel outputs, 517
lists, 84
reports in page structure view, 134
single object, 195
text, 137
values in map legends, 126

G

Gauge Chart, 537
gauge charts, 346

Gauge Labels, 537, 588
Gauge Numerical Axis, 537
Gauge Palette, 588
Generated Prompt, 537
generated prompts, 201
Generated SQL/MDX, 588
Global Customer Services Web site, 38
global filters in dashboards, 199
graphs, *See* charts
Great Outdoors samples, 327, 331
gridlines
    charts, 337
    major, 105
    minor, 105
Gridlines, 589
grouped lists
    subtotals in, 318
grouped reports have incorrect running totals, 323
grouping
    data, 80
Grouping & Sorting, 589
Grouping Type, 589
groups
    definition, 656
Group Size (digits), 637
Group Span, 589
group spans
    setting, 81
guidelines
    laying out reports, 134

H

Has Fact Cells, 589
headers
    adding, 135
    formatting, 135
Heading Level, 589
Height (px), 590
help
    getting, 38
Hide Adornments, 590
Hide Text, 590
hiding
    objects, 261
hierarchies, 64
    definition, 656
    inserting, 170
Hierarchy, 590
Hierarchy Set, 538
highlighting
    data, 257
Hole Size (%), 590
Horizontal Alignment, 591
horizontal charts, *See* bar charts
horizontal pagination, 158
    specifying the number style, 157
HRESULT, 319
HTML, 591
HTML format
    producing a report, 66
HTML Item, 538
HTML items
    inserting, 143
HTML Source Variable, 591
Hyperlink, 538
Hyperlink Button, 538
hyperlink buttons
    inserting, 143
    not supported for Microsoft Excel, 520
hyperlinks
    inserting, 143

I

icons, 64
Ignore Data with No Features, 591
ignoring data with no features
    in maps, 122
Image, 539
images
    inserting, 139
    inserting in the background, 139
    Microsoft Excel limitations, 515
Include Zero For Auto Scale, 592
indenting
    data in crosstabs, 91
    objects, 148
information
    finding, 38
inheritance
    objects, 57
Insertable Objects pane, 51, 52

inserting
background images, 139
blocks, 143
calculations, 143
conditional block lists, 143
dates, 143
field sets, 143
hierarchies, 170
HTML items, 143
hyperlink buttons, 143
hyperlinks, 143
images, 139
layout component references, 143
member properties, 170
page numbers, 157
page sets, 152
repeaters, 143
rich text items, 143
row numbers, 143
tables, 143, 147
International Currency Symbol, 637
Intersection (Tuple), 539
intersection of members, 191
Interval Prompt, 539
interval prompts, 201
adding, 204

J

Japanese
Microsoft Excel limitations, 519
Join, 539
joining
nested page sets, 153
join relationships
creating, 214
Join Relationships, 592

K

Key, 539

L

Label, 592, 593
Label Control, 593
labels
custom, 111
specifying, 111, 112
Labels, 593
languages
adding to reports, 265
Last Date, 594
laying out reports, 134
guidelines, 134
layout component cache, 151
Layout Component Reference, 540
layout component references
inserting, 143
layouts
adding, 264
definition, 657
description, 54
pages, 54
relating to queries, 214
report objects, 55
Left Position (px), 594
Legend, 540, 594
Legend Label, 594
legends
charts, 337
customizing, 113
in maps, 124, 125, 126
positioning, 113
Legend Title, 540, 594
Level, 540, 595
level hierarchies, 167
Level Hierarchy, 541
Level Indentation, 595
levels, 64, 167
definition, 657
member unique names, 270
Level Set, 541
LFA.dll file, 235
limitations
Microsoft Excel output, 515
Line, 541, 595
linear scales
showing, 105
line charts, 94, 341
lines
changing in charts, 105
Line Style, 595
Line Styles, 595

Line Weight (pt), 596
List, 541
List Cell, 541
List Column, 542
List Column Body, 542
List Columns, 542
List Columns Body Style, 543
List Columns Title Style, 543
List Column Title, 543
List Footer, 543
list footers
  adding, 135
List Header, 544
list headers
  adding, 135
list of data items
  specifying for an object, 197
List Page Footer, 544
List Page Header, 544
List Row, 544
List Row Cells Style, 544
lists, 79
  applying table styles, 147
  converting to crosstabs, 91
  converting to repeaters, 130
  creating, 285
  formatting, 84
  grouping data, 80
  headers and footers, 135
  single-edge crosstabs, 87
lists, grouped
  subtotals in, 318
literal strings
  in expressions, 359
live data, 61
locales
  definition, 657
  locale-sensitive properties, 195
logarithmic scales
  showing, 105

M

mailing labels
  creating, 131
managing
  changes in packages, 242
  reports, 77
Mantissa (digits), 637
Map, 545
Map & Layers, 596
Map Drills, 596
Map Location, 545
Map Refinement Location, 545
maps, 119
  aliases, 122
  changing the size of points, 300
  creating, 120, 298
  customizing, 126
  dictionary property, 122
  drill through, 128
  footers, 125
  ignoring data with no features, 122
  legends, 124
  matching data values, 122, 298
  Microsoft Excel limitations, 519
  notes, 126
  parts of a map report, 120
  point layers, 123, 299
  region layers, 121
  resizing, 161
  size of points, 124
  subtitles, 125
  titles, 125
Margin, 596
margins
  setting, 148
Marker, 546
Marker Color, 596
Marker Label, 596
markers
  charts, 106
Markers, 597
Marker Text Location, 597
master detail relationships
  creating, 282
  linking members from two dimensional data sources, 283
  using to join page sets, 153
Master Detail Relationships, 597
Maximum Characters, 597
Maximum Execution Time, 597
Maximum Hotspots, 598

Maximum No. of Digits, 637
Maximum Rows Retrieved, 598
Maximum Size (pt), 598
Maximum Tables, 598
Maximum Text Blob Characters, 599
Maximum Truncation Characters, 599
Maximum Value, 599
MDX, 546, 599
    adding your own, 228
    converting to, 227
    definition, 657
    editing, 227
    viewing, 227
    working with, 227
Measure, 546
measures, 64
    creating calculations, 190
    definition, 657
    specifying default, 198
Member, 546, 599
member calculations, 102
Member Children Set, 547
Member Hierarchy, 547
member intersections, 191
Member Offset (%), 600
member properties, 167
    inserting, 170
Member Property, 547
members, 64
    creating calculations, 190
    definition, 657
    drill-through access, 269
    inserting properties, 170
    linking from two dimensional data sources to create a master detail relationship, 283
Members, 600
Member Set, 547
member sets
    aggregating, 188
    creating, 169, 281
members folders, 167
member unique names, 360
    relational metadata, 270
metric range charts, 346
Metrics Range Chart, 548
Metric Studio applications
    definition, 657
Metric Studio Diagram, 548
Microsoft Excel
    report limitations, 515
Microsoft Excel format
    producing a report, 72
Microsoft Excel limitations
    nested report objects, 519
Minimum No. of Digits, 637
Minimum Size (pt), 600
Minimum Value, 600
Minor Gridlines, 600
missing items in a report or analysis, 321
Missing Value Characters, 637
missing values
    dividing by zero, 317
model
    aggregation properties, 183
model-based drill-through access
    using scope, 272
modeling
    sample models and packages, 335
models, 64
    definition, 657
    using design filters, 66
modifying
    prompts, 207
    properties in page structure view, 134
MOLAP data
    creating burst reports, 251
moving
    crosstab nodes, 86
Multi-Line, 601
multimedia files
    adding, 143
multiple-fact queries
    filters, 215
multiple-page reports, 151
multiple values
    selecting in prompts, 208
Multi-Select, 601
mun, 321
MUN, *See* member unique names

N

Name, 601, 602
Named Set, 548
names
  updating references, 242
native SQL, 227
Negative Column Color, 602
Negative Pattern, 637
Negative Sign Position, 637
Negative Sign Symbol, 638
nested report objects
  Microsoft Excel limitations, 519
nesting
  data in crosstabs, 88
No. of Decimal Places, 638
no data
  specify what appears for a data container, 199
No Data Contents, 199, 603
No Data Features Size (pt), 603
Node Coordinate, 603
Note, 548
Note Border, 603
Note Content, 548
notes
  adding to charts, 107
  in maps, 126
  maps, 126
Notes, 603
number data format
  Microsoft Excel limitations, 519
number of decimals
  rounding mode used by Cognos 8, 195
Number of Regression Lines, 604
Numbers Only, 604
Number Style, 604
Numerical Axis, 549, 604
Numeric Overflow Characters, 638
Numeric Value, 604

O

objects
  adding, 55
  aligning, 147
  as containers, 56
  finding in reports, 57
  formatting, 195
  hiding, 261
  hierarchies, 57
  indenting, 148
  inheritance, 57
  inserting, 143
  reusing, 149
  setting properties, 158
  showing, 261
  types, 56
  updating, 151
OLAP
  data sources, 167, 360
opening
  Analysis Studio analyses, 236
  Query Studio reports, 236
  reports locally, 235
options
  setting, 57
ORA-00907 error, 321
Oracle
  9.2 data source, 321
Ordinal Axis, 549, 605
Outer Join Allowed, 605
overflow error in crosstab report, 321
Override Dimension Info, 605
Overrides, 605

P

packages, 64, 167
  changing connections, 243
  definition, 658
  drilling-through, 272
  managing changes, 242
  referencing items in child queries, 226
  refreshing, 63
  specifying, 62
padding
  applying, 148
Padding, 605
Padding Character, 638
Page, 549
Page Body, 549
page breaks, 158
  controlling, 164
page breaks, *See* page sets

Page Break Text, 550
Page Explorer, 52
Page Footer, 550
Page Header, 550
page layers
creating, 181
Page Number, 550
page numbering, 158
page numbers
inserting, 157
specifying the number style, 157
pages, 54
adding, 151
associating queries, 151
not rendering when there is no data, 70
Page Set, 550
page sets
definition, 658
inserting, 152
joining nested page sets, 153
page structure view, 134
pagination, 157, 158
Pagination, 606
Palette, 606
palettes
charts, 99, 100
conditional, 117
Parameter, 606
parameters
for prompts, 199, 205
using in master detail relationships, 282
Pareto Chart, 551
Pareto charts, 340
parse errors when opening or running an upgraded report, 321
Pattern, 638
patterns
using to format data, 641
PDF page options
setting, 73
PDF Page Setup, 606
PDFs
producing a report, 66
Percentage Symbol, 638
Percentile, 607
Percent of Axis, 607
Percent Scale (integer), 638
Performance Pattern, 607
Pie Chart, 551
pie charts, 339
limitations for Microsoft Excel outputs, 516
Pie Labels, 551, 607
planning
reports, 49
point charts, 344
Point Color, 607
Point Color Legend Title, 552
Point Layer, 552
point layers, 123
Point Measure, 552
Point Shape, 607, 608
Point Size (pt), 608
Point Size Legend Title, 552
Point Size Measure, 552
points layers, 124
Polar Chart, 553
polar charts, 345
Polynomial Exponent, 608
position
calculations, 45
Position, 608
Position type, 608, 609
Positive Column Color, 609
PowerCube drill through does not work, 323
PowerPlay reports
setting up drill through to Cognos 8, 653
Pre-populate, 609
Pre-populate Levels, 609
presentation
changing in charts, 293
Pre-Sort, 609
printing copyright material, 38
Processing, 610
Professional authoring mode, 61
profiles
changing, 62
Progressive Axis, 610
Progressive Chart, 553
progressive column charts, 340
Projection List, 610
Prompt Button, 553

prompt buttons
adding, 211, 212
prompt controls
changing interfaces, 208
date and time prompts, 201
date prompts, 201
generated prompts, 201
interval prompts, 201
select and search prompts, 201
text box prompts, 201
time prompts, 201
tree prompts, 201
value prompts, 201
See Also prompts
prompting, 200
prompt pages
creating your own, 201
prompts
adding, 199
adding prompt buttons, 211
cascading, 212
changing prompt control interfaces, 208
creating, 201
creating in a report page, 206
creating using query macros, 231
definition, 658
interval, 204
modifying, 207
pages, 200
parameters, 199, 205
required, 208
requiring user input, 208
selecting multiple values, 208
showing and hiding status, 209
specifying default selections, 210
specifying values, 210
tree, 212
using the Build Prompt Page tool, 200
properties
charts, 96
locale-sensitive, 195
Properties, 610
Properties pane, 51
properties panes
definition, 658
Property, 610
Push To Bottom, 611

Q

QE-DEF-0288 error, 320
quadrant charts, 345
queries, 55
associating to pages, 151
definition, 658
master detail,, 282
referencing package items in child queries, 226
relating to layouts, 214
working with, 213
working with dimensional data sources, 214
working with MDX, 227
working with SQL, 227
See master detail relationships
Query, 554, 611
query context
defining in charts, 115
Query Explorer, 52, 213
query items
definition, 658
query macros
adding, 231
using to create prompts, 231
Query Operation, 554
Query Reference, 554
Query Studio
creating templates in Report Studio, 75
Query Studio reports
opening, 236
query subjects
definition, 658
quick tours
using, 38
quotation marks
in literal strings, 359

R

Radar Chart, 554
radar charts, 343
Radar Type, 611
Radial Axis, 554, 611
Radial Measure, 555
Range, 611

Range Label, 612
reference
  package items in child queries, 226
referenced objects
  changing, 150
  updating, 151
referencing
  data items, 197
refreshing
  packages, 63
Region Color Legend Title, 555
Region Layer, 555
region layers, 121
Region Measure, 555
Regression Line, 555, 612
regression lines, 105
Regression Type, 612
related documentation, 37
relational-to-cube drill through does not work, 324
relationships
  creating joins, 214
Relative Alignment, 612
removing
  section headings, 181
renaming
  rows or columns, 156
Render Fact Cells, 612
Render Page when Empty, 70, 613
Render Variable, 613
reordering
  rows and columns, 155
Repeater, 556
Repeater Direction, 613
repeaters, 130
  converting from lists, 130
  definition, 658
  inserting, 143
Repeater Table, 556
Repeater Table Cell, 556
repeater tables
  definition, 659
Report Expression, 613, 614, 615
reporting and analysis
  data handling, 40
  table of contents in reports, 40
ReportNet
  preserving report appearance, 164
  report does not retain original look, 322
report objects
  as containers, 56
  hierarchy, 57
  locking and unlocking, 56
  types, 56
  working with, 56
report outputs
  definition, 659
report properties
  changing, 164
reports
  accessible, 75
  adding colors, 138
  basic structure, 54
  building using your own MDX, 228
  building using your own SQL, 228
  bursting, 246
  chart backgrounds appear gray, 318
  chart displays every second label, 318
  chart labels overwrite each other, 318
  combining,, 282
  controlling rows per page, 70
  converting to templates, 74
  copying to the clipboard, 236
  creating, 62
  creating for Cognos 8 Go! Office, 76
  creating multiple-page, 301
  creating prompts in a report page, 206
  creating section headings, 180
  creating sections for dimensional data, 181
  defining conditions, 254
  definition, 659
  drill-through, 273
  drill-through text, 276
  drill-up/drill-down, 279
  finding objects, 57
  formats, 66
  formatting changes, 322
  guidelines for layouts, 134
  laying out, 134
  managing, 77
  multilingual, 265
  not rendering pages when there is no data, 70

opening and saving locally, 235
opening from the clipboard, 236
pages, 54
planning, 49
producing in CSV format, 71
producing in Microsoft Excel format, 72
producing in XML format, 73
removing section headings, 181
reusing styles, 139
running, 66
running against dimensional data sources, 69
running against SAP BW, 69
running in Report Studio, 66
saving in Report Studio, 66
setting PDF page options, 73
summaries do not correspond to visible members, 319
upgrading, 313
validating, 66
See master detail relationships
report samples, 327
report specifications
creating, 62
definition, 659
showing for a selected object, 78
viewing, 78
Report Studio
changing default classes, 164
Insertable Objects pane, 51, 52
planning reports, 49
Properties pane, 51
saving reports, 66
setting options, 57
user interface, 51
visual aids, 52
work area, 52
report types, 79
charts, 94
crosstab, 85
lists, 79
repeaters, 130
Required, 615
required prompts, 208
resizing
charts, 161
maps, 161
reusing
objects, 149
styles, 139
Rich Text Item, 556
rich text items
inserting, 143
supported elements, 145
Right Position (px), 615
Rollup Aggregate Function, 615
Rollup Processing, 616
Root Members Only, 616
Rotate Values, 616
rounding mode
used by Cognos 8 for number of decimals, 195
Row Coordinate, 616
Row Number, 557
row numbers
inserting, 143
rows
changing order, 155
renaming, 156
swapping, 157
rows per page
controlling, 70
Rows Per Page, 66, 617
RSV-SRV-0025, 319
running reports, 66
against dimensional data sources, 69
against SAP BW, 69
cannot find database, 320
ORA-00907 error, 321
running totals
dimensional functions, 189
running totals incorrect, 323
run options
setting, 66

S

Safari browser
drill-through links not active, 323
samples, 327
general employee information, 334
general financial information, 332
general sales and marketing information, 334
GO data warehouse, 335
GO Sales transactional database, 336

Great Outdoors, 331
Great Outdoors company structure, 332
model design, 335
SAP BW data sources
creating burst reports against, 251
running reports against, 69
SAP variable properties
unsupported, 70
saving
reports, 66
reports locally, 235
Scale, 617, 638
Scale Interval, 617
scales
changing interval, 105
linear, 105
logarithmic, 105
Scatter Chart, 557
scatter charts, 343
scope
setting in summary filters, 172
using in drill-through access, 272
Scope, 617
SDK, *See* Cognos 8 Software Development Kit
searching
dimensions, 171
searching for values
unexpected results, 359
searching reports, *See* finding objects
Secondary Group Size (digits), 639
section headings
creating, 180
removing, 181
sections
creating for dimensional data, 181
sector charts, *See* pie charts
Secure Socket Layer
Microsoft Excel limitations, 518
Security Error Characters, 639
Select & Search Prompt, 557
select and search prompts, 201
Select UI, 617
Separator, 617
Series Color, 618
Set Expression, 557
Set Operation, 618
sets
creating, 169
Set Sorting, 618
setting
burst options, 249
margins, 148
object properties, 158
run options, 66
Show Caption, 618
Show Data Range in Legend, 618
Show Features with No Data, 619
Show Feelers, 619
showing
objects, 261
the report specification, 78
Show Legend Values, 619
single-edge crosstabs
creating, 87
Singleton, 557
singletons
inserting, 196
Size & Overflow, 619
Size and Overflow, 161
Size Legend Title, 620
slicer filters, 174
prompting, 200
Slicer Member Set, 558
slicers
changed features, 44
defining, 172
solve order, 191
combined with calculated rollup aggregation function, 192
Solve Order, 620
sorting
data, 177
multiple columns, 179
Sorting, 620
source trees
customizing, 66
Source Type, 620, 621
Spacing & Breaking, 621
Spacing and Breaking, 160
special values, 44
specifying
default measures, 198

fonts, 138
labels, 111, 112
packages, 62
spider charts, *See* radar charts
Spider Effects, 621
SQL, 558, 621
adding your own, 228
Cognos, 227
converting to, 227
editing, 227
first-rows optimization, 229
native, 227
viewing, 227
working with, 227
SQL join syntax, 622
SQL Syntax, 622
SSL, See Secure Socket Layer, 518
stacked charts, 348
standard charts, 347
Standard Deviations, 622
star charts, *See* radar charts
Static Choices, 622
status
showing and hiding in prompts, 209
strings
concatenating, 361
styles
reusing, 139
style sheets
definition, 659
Style Variable, 622
Subtitle, 623
subtotals
in grouped lists, 318
summaries
adding, 182
automatic, 351
average, 351
calculated, 192, 351
count, 351
count distinct, 351
custom, 351
definition, 659
maximum, 351
median, 351
minimum, 351
none, 351
not applicable, 351
standard deviation, 351
summarize, 351
total, 351
variance, 351
summaries in reports do not correspond to visible members, 319
summarizing
data, 182
Summary Filter, 558
Suppress, 623
suppressing
data, 192
swapping
columns and rows, 157

T

Table, 558
Table Cell, 558
table of contents
creating, 140
Table of Contents, 559, 623
Table of Contents Entry, 559
table of contents in reports, 40
Table of Contents Name, 623
Table Properties, 624
Table Row, 559
tables
applying styles, 147
inserting, 143, 147
setting properties, 163
table widths
Microsoft Excel limitations, 518
tabular data
viewing, 66
Target Color, 624
Target Marker, 624
Target Marker Border Color, 624
Target Marker Position, 624
Target Measure, 559
Target Range (%), 624
target reports
creating for drill through from Cognos Series 7, 654
template
background color does not appear, 317

templates
choosing, 63
converting from reports, 74
creating, 74
definition, 659
text
adding, 137
formatting, 137
Text, 625
Text Box Prompt, 559
text box prompts, 201
Text Flow & Justification, 625
Text Flow and Justification, 160
Text Item, 560
Text Source Variable, 625
Thousands Separator, 639
Time, 560
Time Prompt, 560
time prompts, 201
Time Separator, 639
Time Style, 639
Time Unit, 639
Title, 625
titles
chart axis, 115
in maps, 126
TOC
creating, 140
Tolerance Color, 626
Tolerance Label, 626
Tolerance Measure, 560
tooltips
in maps, 126
Tooltips, 626
Top Position (px), 626
Total Column, 561, 626
Total Column Color, 626
Total Column Label, 561
Tree Prompt, 561
tree prompts, 201
controlling the data that appears, 212
trees
customizing, 66
trend lines, 105
troubleshooting, 319
Truncation, 627
Truncation Text, 627
tuples, 191
Type, 627

U

UDA-SQL-0114, 319
UDA-SQL-0206, 319
ungrouping
data, 80
union
combining sets in charts, 291
units of measure, 193
units of measure notation, 70
unsupported SAP variable properties, 70
updating
copied objects, 151
name references, 242
referenced objects, 151
upgrade messages
removing, 313
upgrading
changes to product behaviors, 44, 315
report will not open, 321
upgrading reports
members not suppressed for SAP BW, 315
removing upgrade messages, 313
thousands separators, 315
Upper Range Skew (%), 628
URL, 628
URL Source Variable, 628
Usage, 628
Use 1.x Behavior, 628
Use Detail Value on Page, 629
Use for Parameter Info, 629
Use Local Cache, 629
user input
requiring in prompts, 208
user profiles, 61
changing, 62
Use Same Range For All Instances, 629
Use SQL With Clause, 629
Use Thousands Separator, 630, 639
Use Value, 630
using
model design filters, 66

V

validating
reports, 66
Value Location, 630
Value Prompt, 561
value prompts, 201
Value Representation, 630
values
customizing in charts, 289
showing in charts, 289
special, 44
specifying in prompts, 210
Values, 630, 631
Value Type, 631
Variable, 561
variables
adding, 259
Vertical Alignment, 631
vertical bar charts, *See* column charts
viewing
MDX, 227
SQL, 227
tabular data, 66
the report specification, 78
viewing reports
incorrect running totals in grouped reports, 323
Visible, 631
visual aids, 52
Visual Angle, 632

W

waterfall charts, *See* progressive column charts
Web browsers
configuring, 61
Web-safe
color palettes, 138
White Space, 632
Width (px), 632
widths
Microsoft Excel limitations, 518
work areas, 52
definition, 659
working
with dimensions, 167
with report objects, 56

X

X Axis, 562, 632
X Axis Measure, 562
XLS format
limitations, 515
XML
producing a report, 73
XY charts, *See* scatter charts

Y

Y1 Axis, 562, 633
Y2 Axis, 562, 633
Y2 Axis Position, 633
Y Axis, 563, 633
Y Axis Measure, 563

Z

Z Axis, 563
Z Axis Measure, 564
Zero Value Characters, 639